



Themen für die mündliche RDP 2019 als PDF-Download

Themenpool Matura 2019 - 8BI

- 1) Zahlensysteme
- 2) Schaltalgebra
- 3) Audiotbearbeitung
- 4) Bild- und Videobearbeitung
- 5) Betriebssysteme Windows und Linux
- 6) Hardware - Grundlagen, Vertiefung und Entwicklung
- 7) Textverarbeitung - MS Word
- 8) Tabellenkalkulation - MS Excel
- 9) Datenbanksysteme - ER-Modell, Relationenmodell, SQL
- 10) Satzsystem LaTeX
- 11) HTML - Grundlagen und Vertiefung - CSS
- 12) PHP - Grundlagen und Vertiefung
- 13) PHP/MySQL Datenbanken
- 14) Grundlagen des Programmierens in C++
- 15) Grundlagen des visuellen (objektorientierten) Programmierens
- 16) Strukturen - Verkettete Listen, binäre Bäume, Rekursionen
- 17) Contentmanagementsysteme
- 18) Netzwerke - Filius
- mündliche RDP - Umfeldfragen

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura



Last update: **2019/05/14 16:04**

1) Zahlensysteme in der Informatik

- [Kurzüberblick - Geschichtliche Entwicklung des Zählens](#)
- [Skriptum zur Einführung](#)
- [Informationseinheiten in der Informatik](#)

-
- [1.01\) Dualsystem](#)
 - [1.02\) Hexadezimalsystem](#)
 - [1.03\) Oktalsystem](#)
 - [1.04\) Umrechnungen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:01



Last update: **2019/04/01 07:42**

1.01) Dualsystem

Das **duale Zahlensystem** - auch **Dualsystem** oder **Binärsystem** genannt - besteht aus **2 Ziffern**, gekennzeichnet durch 0 und 1. Man benötigt dieses Zahlensystem in der Informatik, da sich mit technischen Bauteilen sehr leicht die Zustände AN und AUS erzeugen lassen können. Diese Zahlen können entsprechend unserem „normalen“ Dezimalsystem verwendet werden. Man kann sie addieren, subtrahieren, multiplizieren und dividieren. Da sie sich also kaum vom „normalen“ Rechnen unterscheiden, eignen sie sich hervorragend, um in der EDV eingesetzt zu werden.

Zählen im Dualsystem

Auch hier beginnen wir mit 0 und zählen dann 1. Leider haben wir nur 2 Zahlen, also gehen uns hier die Zahlen schnell aus. Wir machen es jetzt aber genau wie im Dezimalsystem und nehmen eine Stelle dazu. Nach 0 und 1 kommen dann also 10 und 11. Wieder reichen die Stellen nicht! Also noch eine dazu: 100, 101, 110, 111, usw.

Zählen von 0 bis 15 im Dezimal- und Dualsystem

Dezimal	Dual
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Eine andere Schreibweise

Man kann Zahlen auch anhand ihrer Basis darstellen. Im Dezimalsystem haben wir 10 Zahlen zur Verfügung, von 0 bis 9. Mit 2 Stellen können wir also $10 * 10 = 100$ Zahlen darstellen. 100 Zahlen? Aber 100 hat doch drei Stellen! Dieser Einwand stimmt. Da wir jedoch mit der Zahl 0 beginnen, ist 0 die 1. Zahl, 1 die 2. Zahl, ... 98 die 99. Zahl und 99 die 100. Zahl.

Mit 3 Stellen können wir $10 * 10 * 10 = 1000$ Zahlen darstellen. Jede Stelle entspricht einer 10-er Potenz.

An einem einfachen Beispiel versuche ich diesen Sachverhalt zu erklären.

Wir nehmen dazu die Zahl 372 und schreiben sie als kleine Rechnung auf:

$$372 = 3 \cdot 100 + 7 \cdot 10 + 2 \cdot 1.$$

Das kann man jetzt noch anders darstellen als:

$$3 \cdot 10^2 + 7 \cdot 10^1 + 2 \cdot 10^0.$$

Auf diese Weise kann man jetzt alle anderen Zahlen auch darstellen:

$$6574 = 6 \cdot 10^3 + 5 \cdot 10^2 + 7 \cdot 10^1 + 4 \cdot 10^0$$

$$12032 = 1 \cdot 10^4 + 2 \cdot 10^3 + 0 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0$$

Verwendung der Potenzschreibweise bei binären Zahlen

Wendet man die Potenzschreibweise bei binären Zahlen an, so muss man eine andere Basis wählen. Es gibt ja nur 2 verschiedene Ziffern, 0 und 1. Also nehmen wir als Basis 2.

Die Zahl 1011 schreibt sich dann als

$$1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Die Umrechnung von dezimalen in binäre Zahlen

Bei der Umrechnung der Dezimalzahlen verwenden wir die „Division mit Rest“ aus der Grundschule. Wir teilen die Zahl solange durch 2, bis als Ergebnis 0 herauskommt und merken uns dabei den Rest. Als Beispiel sollen die Zahlen 13 und 14 dienen.

$$13 / 2 = 6 \text{ Rest } 1$$

$$6 / 2 = 3 \text{ Rest } 0$$

$$3 / 2 = 1 \text{ Rest } 1$$

$$1 / 2 = 0 \text{ Rest } 1$$

Die Reste von unten nach oben aneinander gereiht ergeben dann die Dualzahl 1101.

$$14 / 2 = 7 \text{ Rest } 0$$

$$7 / 2 = 3 \text{ Rest } 1$$

$$3 / 2 = 1 \text{ Rest } 1$$

$$1 / 2 = 0 \text{ Rest } 1$$

Hieraus ergibt sich dann die Dualzahl 1110.

Addition von Dualzahlen

Einige erinnern sich vielleicht noch an die Addition von Zahlen wie wir sie in der Grundschule gelernt haben. Wir schreiben die Zahlen untereinander und addieren sie Stelle für Stelle. Dabei beginnen wir mit der letzten Stelle und arbeiten uns langsam nach vorne durch. Die gleiche Vorgehensweise

benutzen wir nun auch wenn wir Dualzahlen addieren wollen.

Die Addition funktioniert wie bei der Addition von Dezimalzahlen:

```

0111
+0100
====
1011

```

Addition mit Überlauf:

```

1111
+0100
====
10011

```

Es gelten die Regeln $0+0=0$, $1+0=1$, $0+1=1$, $1+1=0$ Übertrag 1. Im Prinzip also nichts neues. Addiert man im Dezimalsystem 2 Zahlen so kommt bei $5+5$ auch 0 Übertrag 1 heraus. Der Übertrag wird bei beiden System jeweils voran gestellt. Also gilt im Dualsystem $1+1=10$.

Vorsicht Überlauf!

Die Addition ist im Prinzip problemlos. Es gibt allerdings einen Haken an der Sache: Die Addition funktioniert nur innerhalb eines bestimmten Wertebereiches. Woran liegt das? In der Realität können wir beliebig grosse Zahlen darstellen. Das geht leider nicht in der Informatik. Wir haben nur einen begrenzten Raum bzw. Speicherplatz zur Verfügung. Wir müssen aus diesem Grund den Speicherbereich einschränken (siehe was ist ein Byte), in unserem Beispiel nehmen wir als Speicherplatz ein Byte. Normalerweise verwendet man zur Addition von ganzen Zahlen, einen deutlich größeren Wertebereich, aber um einen überschaubaren Rahmen zu haben, begrenzen wir uns absichtlich auf ein Byte. Unsere größte darstellbare Zahl ist die 11111111, also dezimal 255. Was passiert nun, wenn wir eine 00000001, also 1, addieren? Das verrückte ist: es kommt 00000000, also 0 heraus. Da bekanntlich $1+1=0$ Übertrag 1 gibt, bekommt man als Ergebnis in Wirklichkeit nicht 00000000 sondern 00000000 Übertrag 1. Dieser letzte Übertrag kann jedoch nicht mehr gespeichert werden und wird deshalb einfach ersatzlos gestrichen. Es ergibt sich daraus jedoch auch eine gewisse Logik. Durch meinen beschränkten Wertebereich komme ich irgendwann an meine obere Grenze. Bei der Addition von 1 fängt dann jedoch der Wertebereich wieder von vorne an, ich bin jetzt an der unteren Grenze, man durchläuft dann wieder den Bereich bis zur oberen Grenze, usw. Wenn wir also immer und immer wieder 1 addieren zählen wir unendlich oft von 0 bis 255, dann wieder 0 bis 255 usw.

Subtraktion von Dualzahlen

Drei Schritte zu Subtraktion

Hier kommen wir mit unserer normalen Schulmathematik nicht mehr weiter. Bevor wir uns mit dem komplizierten „Warum ist das denn so?“ beschäftigen, merken wir uns erst einmal den Mechanismus. Die Subtraktion von binären Zahlen wird durch die **Addition des Zweierkomplementes**

durchgeführt. Zur Erklärung beginnen wir im ersten Schritt mit dem **Einerkomplement**, dann schauen wir im zweiten Schritt was das **Zweierkomplement** ist und dann kommen wir im letzten Schritt zur **Subtraktion**.

Das Einerkomplement

Was ist das Komplement von Dualzahlen? Man bildet das sogenannte Einerkomplement, indem man jede Zahl durch ihr Gegenteil ersetzt, also die 0 durch die 1 und die 1 durch die 0.

01011010 wird zu 10100101 11101101 wird zu 00010010

Das Zweierkomplement

Das Zweierkomplement entspricht dem Einerkomplement, nur wird zusätzlich noch 00000001 addiert.

01011010 wird im Einerkomplement zu 10100101 im Zweierkomplement zu 10100110 11101101 wird im Einerkomplement zu 00010010 im Zweierkomplement zu 00010011

Die Subtraktion von Dualzahlen

Der Satz lautet: Die Subtraktion von 2 Zahlen erfolgt durch die Addition des Zweierkomplementes. Als konkretes Beispiel nehmen wir dazu die Rechnung $14-9=5$.

!!WICHTIG:!!

Die restlichen Ziffern müssen immer mit 0 aufgefüllt werden.

9 ist im Dualsystem 00001001.

Das Einerkomplement zu 00001001 ist 11110110.

Das Zweierkomplement 11110111.

Dies addieren wir nun zu 14 also 00001110.

```
00001110
+11110111
=====
00000101
```

Auch hier wäre die richtige Zahl eigentlich 00000101 Übertrag 1, da wir den Übertrag jedoch nicht speichern können, bleiben wir bei 00000101 was ja der Dezimalzahl 5 entspricht.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:01:01_01



Last update: **2019/04/01 07:41**

1.02) Hexadezimalsystem

Besonders **wichtig** ist in der **Informatik und Digitaltechnik** neben dem Binärsystem auch das **Hexadezimalsystem (Sedezimalsystem)**. Das Hexadezimalsystem verwendet die **Basis 16**, d.h. es gibt **16 verschiedene Ziffern, 0 bis 9** und zusätzlich die Buchstaben **A bis F** (sog. Zahlzeichen; können auch als klein geschrieben werden: a-f).

Mit dem Hexadezimalsystem können auf einfachere und kürzere Weise Binärzahlen notiert werden. Mit einer 4-stelligen Binärzahl (auch als Halbbyte oder Nibble bezeichnet) lassen sich 16 ($2^4 = 16$) verschiedene Zahlen darstellen, und zwar 0 bis 15 (die Null zählt mit!). Da das Hexadezimalsystem die Basis 16 ($= 2^4$) verwendet, reicht eine (!) Hexadezimalzahl aus, um vier Bits (Binärziffern) darzustellen. Mit zwei Hexadezimalzahlen kann ein Byte (8 Bits) angeschrieben werden.

Gegenüberstellung Hexadezimal-, Binär- und Dezimalsystem

Hex	Binär	Dezimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Um eindeutig darauf hinzuweisen, dass es sich um eine Hexadezimalzahl handelt, kann ebenso wie in anderen Zahlensystemen die Basis tiefgestellt dazu geschrieben werden, z.B. **3F₁₆** ($= 63_{10}$ dezimal) oder **93₁₆** ($= 147_{10}$ dezimal). Es sind aber auch andere Schreibweisen üblich:

a) Vorangestelltes **0x (Prefix)**, z.B. **0x93**. Diese Notation wird in Programmiersprachen mit C-ähnlicher-Syntax verwendet.

b) Nachgestelltes **h (Postfix)**, z.B. **93h**. Letztere Schreibweise ist besonders in der Technik gebräuchlich.

Umrechnung vom Dezimal- ins Hexadezimalsystem

Die Umrechnung funktioniert ähnlich der Umrechnung von Dezimal- zu Binärzahlen (s.o.). Nun muss aber, statt durch 2, durch 16 dividiert werden. Die Reste werden genauso von rechts nach links angeschrieben und geben, wenn das Ergebnis der Ganzzahldivision 0 ist, das Endergebnis.

Beispiel: Die Dezimalzahl **304**₁₀ soll in eine Hexadezimalzahl umgewandelt werden.

```
304 / 16 = 19 => 0 Rest
 19 / 16 =  1 => 3 Rest
  1 / 16 =  0 => 1 Rest
```

Für das Endergebnis werden jetzt die Reste **von unten nach oben gelesen**.

Somit ergibt sich ein Endergebnis von 130₁₆, das entspricht der Dezimalzahl **304**₁₀.

Umrechnung vom Hexadezimal- ins Dezimalsystem

Die Umrechnung vom Hexadezimal- ins Dezimalsystem kann genauso wie oben von Binär→Dezimal demonstriert, erfolgen. Die einzelnen Ziffern werden mit dem jeweiligen Stellenwert (16^n , wobei $n = 0, 1, 2, \dots$) multipliziert und die jeweiligen Ergebnisse aufsummiert. Das folgende Beispiel demonstriert dies anhand der Hexadezimalzahl 130₁₆:

```
0 * 16^0 = 0
3 * 16^1 = 48
1 * 16^2 = 256
-----
          = 304
```

Als Ergebnis erhalten wir 304 dezimal, womit die Probe - zur vorigen Rechnung in die umgekehrte Richtung - erfolgreich war. 304₁₀ entspricht 130₁₆. Diese Antwort hätte in der Praxis natürlich auch ein wissenschaftlicher Taschenrechner geliefert. 😊 Es reicht dazu sogar der **Windows-Rechner** (den Sie nur auf die wissenschaftliche Ansicht umstellen müssen) oder unter Linux Programme wie z.B. KCalc.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:01:01_02

Last update: **2019/04/01 07:42**



1.03) Oktalsystem

Das Oktalsystem, auch Achtersystem genannt, verwendet die **Basis 8 (acht)**. Um Zahlen darzustellen, stehen die **Ziffern 0 bis 7** zur Verfügung. Die Bedeutung in der Informatik/Digitaltechnik ergibt sich dadurch, dass sich mit einer **Oktalzahl drei Bits** darstellen lassen. 23 ist 8, somit lassen sich mit 3 Bits 8 verschiedene Möglichkeiten darstellen. Eine Oktalzahl reicht, um diese Information wiederzugeben.

Das Oktalsystem wird hier insbesondere deshalb erwähnt, weil in vielen Programmiersprachen Zahlen auch in Oktalform angegeben werden können. Meist, z.B. in PHP, wird dazu eine 0 (Null) vorangestellt, z.B. 077 für $77_8 (= 63_{10})$.

Umrechnungen erfolgen genauso wie beim Hexadezimalsystem gezeigt. Um die Oktalzahl auszurechnen, die einer best. Dezimalzahl entspricht, dividieren Sie die Dezimalzahl fortlaufend durch 8 und schreiben die Reste von rechts nach links an. In umgekehrter Richtung - von Oktal nach Dezimal - multiplizieren Sie die einzelnen Ziffern mit dem Stellenwert (8^n für $n = 0, 1, 2, \dots$) und addieren die Teilergebnisse.

Umrechnung vom Dezimal- ins Oktalsystem

Die Umrechnung funktioniert ähnlich der Umrechnung von Dezimal- zu Binärzahlen (s.o.). Nun muss aber, statt durch 2, durch 8 dividiert werden. Die Reste werden genauso von rechts nach links angeschrieben und geben, wenn das Ergebnis der Ganzzahlendivision 0 ist, das Endergebnis.

Beispiel: Die Dezimalzahl **304₁₀** soll in eine Hexadezimalzahl umgewandelt werden.

```
304 / 8 = 38 => 0 Rest
 38 / 8 =  4 => 6 Rest
  4 / 8 =  0 => 4 Rest
```

Für das Endergebnis werden jetzt die Reste **von unten nach oben gelesen**.

Somit ergibt sich ein Endergebnis von 460₈, das entspricht der Dezimalzahl **304₁₀**.

Umrechnung vom Oktal- ins Dezimalsystem

Die Umrechnung vom Oktal- ins Dezimalsystem kann genauso wie oben von Binär→Dezimal demonstriert, erfolgen. Die einzelnen Ziffern werden mit dem jeweiligen Stellenwert (8^n , wobei $n = 0, 1, 2, \dots$) multipliziert und die jeweiligen Ergebnisse aufsummiert. Das folgende Beispiel demonstriert dies anhand der Oktalzahl 460₈:

```
0 * 8^0 = 0
6 * 8^1 = 48
4 * 8^2 = 256
-----
      = 304
```

Als Ergebnis erhalten wir 304 dezimal, womit die Probe - zur vorigen Rechnung in die umgekehrte Richtung - erfolgreich war. 304_{10} entspricht 460_8 . Diese Antwort hätte in der Praxis natürlich auch ein wissenschaftlicher Taschenrechner geliefert. 😊 Es reicht dazu sogar der **Windows-Rechner** (den Sie nur auf die wissenschaftliche Ansicht umstellen müssen) oder unter Linux Programme wie z.B. KCalc.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:01:01_03



Last update: **2019/04/01 07:41**

Umrechnungen

Prinzipiell kann man jede Zahl von einem Zahlensystem in ein anderes Zahlensystem umrechnen. Dazu muss man normalerweise die Zahl immer zuerst in das Dezimalsystem und dann in das Ziel-Zahlensystem umrechnen. Ausnahmen gibt es bei Umrechnungen vom Binärsystem in ein anderes Zahlensystem, hier braucht man nicht unbedingt das Dezimalsystem.

Binärsystem \Leftrightarrow Oktalsystem

Binärsystem \Rightarrow Oktalsystem

Wir wissen ja bereits, dass eine Ziffer im Oktalsystem die Ziffern 0-7 annehmen kann und dass man 3 Bits im Dualsystem braucht um 8 verschiedene Zahlenwerte darzustellen.

\Rightarrow Immer 3 Stellen im Binärsystem ergeben eine Ziffer/Stelle im Oktalsystem

Beispiel

Umwandlung der Zahl 010110011111_2 ins Hexadezimalsystem:

010	110	011	111
↓	↓	↓	↓
2	6	3	7

Die Zahl 010110011111_2 entspricht somit der Zahl 2637_8 .

Oktalsystem \Rightarrow Binärsystem

Selbiges gilt auch für die Umrechnung vom Oktalsystem in das Binärsystem.

\Rightarrow Eine Stelle im Oktalsystem entspricht 3 Stellen im Binärsystem

Beispiel

Umwandlung der Zahl 672_8 ins Binärsystem:

6	7	2
↓	↓	↓
110	111	010

Die Zahl 672_8 entspricht somit der Zahl 110111010_2 .

Binärsystem \Leftrightarrow Hexadezimalsystem

Binärsystem \Rightarrow Hexadezimalsystem

Wir wissen ja bereits, dass eine Ziffer im Hexadezimalsystem die Werte 0-15 annehmen kann und dass man 4 Bits im Dualsystem braucht um 16 verschiedene Zahlenwerte darzustellen.

\Rightarrow Immer 4 Stellen im Binärsystem ergeben eine Ziffer/Stelle im Oktalsystem

Beispiel

Umwandlung der Zahl 010110011111_2 ins Oktalsystem:

0101	1001	1111
↓	↓	↓
5	9	F

Die Zahl 010110011111_2 entspricht somit der Zahl $59F_{16}$.

Hexadezimalsystem \Rightarrow Binärsystem

Selbiges gilt auch für die Umrechnung vom Hexadezimalsystem in das Binärsystem.

\Rightarrow Eine Stelle im Hexadezimalsystem entspricht 4 Stellen im Binärsystem

Beispiel

Umwandlung der Zahl 672_{16} ins Binärsystem:

6	7	2
↓	↓	↓
0110	0111	0010

Die Zahl 672_{16} entspricht somit der Zahl 11001110010_2 .

Oktalsystem \Rightarrow Hexadezimalsystem

Will man vom Oktalsystem ins Hexadezimalsystem umrechnen, rechnet man am besten zuerst ins Binärsystem und dann ins Hexadezimalsystem um.

Hexadezimalsystem \Rightarrow Oktalsystem

Will man vom Hexadezimalsystem ins Oktalsystem umrechnen, rechnet man am besten zuerst ins Binärsystem und dann ins Oktalsystem um.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

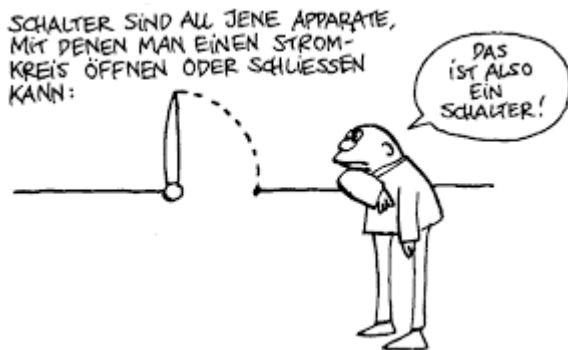
Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:01:01_04



Last update: **2019/04/01 07:42**

2) Schaltalgebra



- 2.01) Grundlagen
- 2.02) Grundsaltungen
- 2.03) Zusammengesetzte Schaltungen
- 2.04) Gesetze der Schaltalgebra
- 2.05) Digitale Rechenschaltungen
- 2.06) Übungen
- 2.07) Übungen
- 2.08) Übungen

Simulator für logische Schaltungen

<https://elektroniktutor.de/swfdatei/gatter.swf>

Das [Adobe Flash Plugin](#) wird benötigt, um diesen Inhalt anzuzeigen.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02

Last update: **2019/05/14 15:39**



2.01) Grundlagen

In der Digitaltechnik sind die Eingangsvariablen so miteinander zu verknüpfen, dass die Ausgangsvariable einen definierten Zustand annimmt, um damit einen Prozess zu steuern. So soll ein Fahrstuhl nur dann fahren, wenn eine Zieletage gewählt wurde, in der er zurzeit nicht steht, seine Tür vollständig geschlossen ist und von ihr nichts eingeklemmt wurde und die Kabine nicht überlastet ist. Man kann durch Kombinieren der beschriebenen digitalen Gatter und Ausprobieren bei einfacheren Aufgaben die eine oder andere Lösung finden. Das eigentliche Ziel ist es, eine wirtschaftliche Digitalschaltung zu entwickeln, die mit einem Minimum gleicher Gattertypen zum Ziel führt.

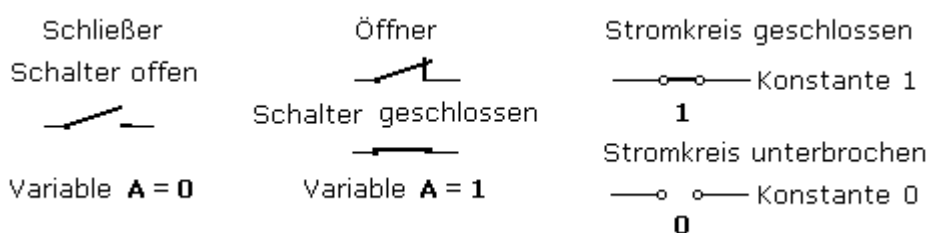
Der englische Mathematiker Georg Boole entwickelte eine Mengenalgebra, die in angepasster Weise als Boolesche Schaltalgebra bei der Problemlösung hilfreich ist. In der Schaltalgebra gibt es Variablen und Konstanten. Die binäre Digitaltechnik kommt mit zwei definierten logischen Zuständen, der 0 und 1 aus.

Konstante

In der Schaltalgebra gibt es nur die zwei konstanten Größen 0 und 1. In der elektronischen Schaltung entspricht eine dauerhaft geschaltete Leitung der Konstanten mit dem Zustand 1. Die dauerhafte Unterbrechung eines Stromkreises steht für die Konstante mit dem Wert 0.

Variable

Veränderbare, schaltbare Eingangsgrößen und davon abhängige veränderliche Ausgangswerte werden als Variable bezeichnet. In der binären Digitaltechnik nehmen sie entweder den Zustand 0 oder 1 an. In elektronischen Schaltungen können sie mit einem Schalter verglichen werden. Der geöffnete Schalter entspricht einer Variablen mit dem Wert 0. Bei geschlossenem Schalter nimmt die Variable den Wert 1 an.



Wir wissen, dass alle Grundrechnungsarten auf eine Addition zurückgeführt werden können. Deshalb ist das Ziel dieses Kapitels eine Maschine simulieren zu können, die addieren kann.

Informationseinheiten

BIT

Die Ziffern 0 und 1 werden physikalischen Zuständen zugeordnet:

0	Strom fließt nicht
1	Strom fließt

0 und 1 werden in der Informatik mit Bit bezeichnet. Ein Bit stellt die kleinste Informationseinheit dar (nur 2 Zustände – 0 und 1).

1 Bit $\Rightarrow 2^1 \Rightarrow 2$ Möglichkeiten

2 Bit $\Rightarrow 2^2 \Rightarrow 4$ Möglichkeiten

3 Bit $\Rightarrow 2^3 \Rightarrow 8$ Möglichkeiten

8 Bit = 1 Byte $\Rightarrow 2^8 \Rightarrow 256$ Möglichkeiten

BYTES

8 Bits sind ein Byte!

1 KB (Kilobyte) = 2^{10} = 1024 Byte

1 MB (Megabyte) = 2^{10} KB = $2^{10} \times 2^{10}$ Byte = 2^{20} Byte = 1048576 Byte

1 GB (Gigabyte) = 2^{10} Byte

1 TB (Terrabyte) = 2^{10} Byte

Darstellung von 0 und 1

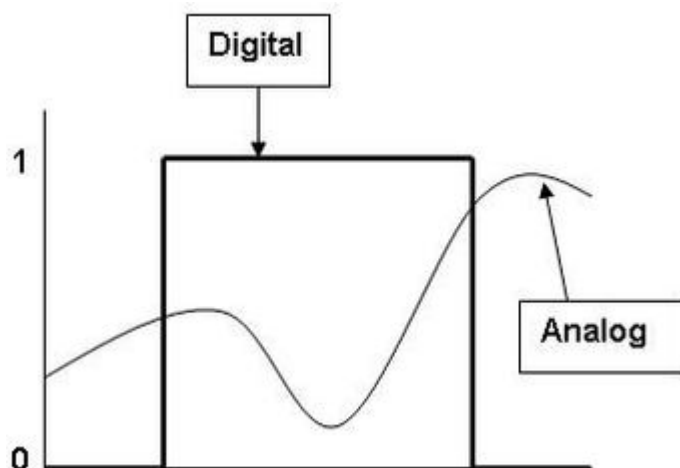
Ein Computer verarbeitet systembedingt Digitalsignale, da nur ausgewertet werden kann, ob eine Spannung anliegt oder nicht bzw. innerhalb eines Definitionsbereichs einen Wert über- oder unterschreitet. Diese zwei Zustände werden auch häufig bezeichnet als:

HIGH und **LOW**

TRUE und **FALSE**

AN und **AUS**

1 und **0**



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_01



Last update: **2019/05/14 15:27**

2.02) Grundsaltungen

UND-Verknüpfung (=Serienschaltung)

Bei der UND-Verknüpfung führt der Ausgang das Signal 1, wenn an beiden Eingängen (a und b) das Signal 1 anliegt.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)	Zeichen (Mathematik)
UND	AND	Konjunktion	$X=A \wedge B$	$X=A*B$

Schaltwerttabelle

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

Schaltfunktion

$$f(a,b)=a \wedge b$$

Schaltsymbol



Bild 5-5: Schaltzeichen für UND-Gatter

elektronische Schaltung

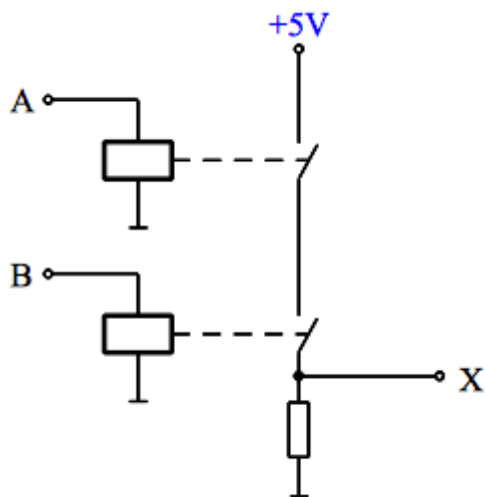


Bild 5-4: UND-Verknüpfung mit Relais

ODER-Verknüpfung (=Parallelschaltung)

Bei der ODER-Verknüpfung führt der Ausgang das Signal 1, wenn an einem der beiden Eingänge das Signal 1 anliegt.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)	Zeichen (Mathematik)
ODER	OR	Disjunktion	$X=A \vee B$	$X=A+B$

Schaltwerttabelle

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Schaltfunktion

$$f(a,b)=a \vee b$$

Schaltsymbol

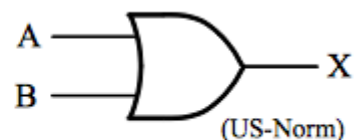
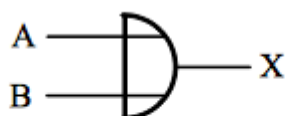
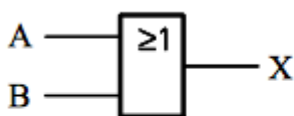


Bild 5-8: Schaltzeichen für ODER-Gatter

elektronische Schaltung

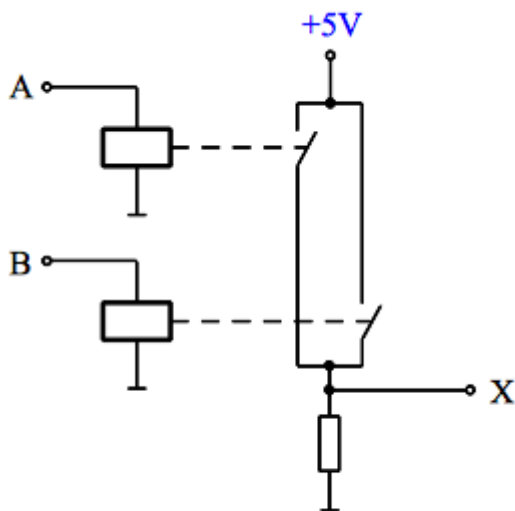


Bild 5-7: ODER-Verknüpfung mit Relais

NICHT-Verknüpfung (=NOT-Schaltung)

Bei einer NICHT-Verknüpfung wird der Ausgang logisch 1, wenn der Eingang logisch 0 ist, und umgekehrt.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)	Zeichen (Mathematik)
NICHT	NOT	Negation	$X = \neg A$	$X = \bar{A}$

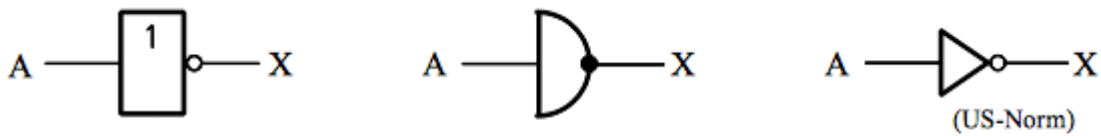
Schaltwerttabelle

a	$\neg a$
0	1
1	0

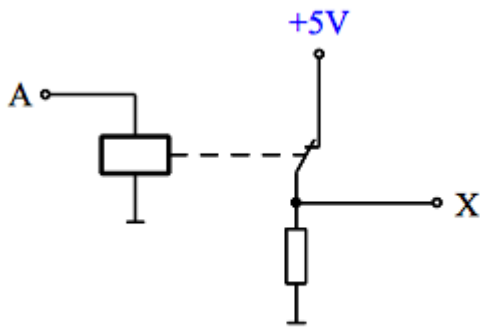
Schaltfunktion

$$f(a) = \neg a$$

Schaltsymbol

**Bild 5-11:** Schaltzeichen für NICHT-Gatter

elektronische Schaltung

**Bild 5-10:** NICHT-Verknüpfung mit Relais

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_02



Last update: **2019/05/14 15:27**

2.03) Zusammengesetzte Schaltungen

Die **Grundverknüpfungen** werden in der Regel miteinander **kombiniert**, sodass die Logik der Schaltung verändert wird. Beispielsweise kann man eine UND- mit einer NICHT-Verknüpfung kombinieren und erhält dadurch eine sogenannte NAND-Verknüpfung (NICHT-UND). Da solche zusammengesetzte Schaltfunktionen sehr häufig verwendet werden, haben sie **eigene Schaltzeichen** bekommen. Mit den folgenden Schaltfunktionen werden die Grundverknüpfungen erweitert.

- NAND
- NOR
- XOR
- XNOR

NAND-Verknüpfung

Eine Kombination aus einem **UND-Gatter mit nachfolgendem NICHT-Gatter ergibt ein NAND-Gatter**. Die Ausgangsvariable Z des UND-Gatters wird durch das NICHT-Gatter negiert und erzeugt die Ausgangsvariable X des NAND-Glieds. Viele logische Funktionen lassen sich durch den Einsatz von NAND-Gattern lösen. Oft steigt dadurch die Anzahl der notwendigen Gatter, mit dem wirtschaftlichen Vorteil nur einen Gattertyp zu verwenden.

Der Ausgangszustand eines NAND-Glieds ergibt 1, wenn nicht alle Eingangszustände 1 sind.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)
NEGIERTES UND	NOT AND	Negative Konjunktion	$Z = \overline{A \wedge B}$

Schaltwerttabelle

a	b	$a \overline{b}$
0	0	1
0	1	1
1	0	1
1	1	0

Schaltfunktion

$$f(a,b) = a \overline{b}$$

Schaltsymbol

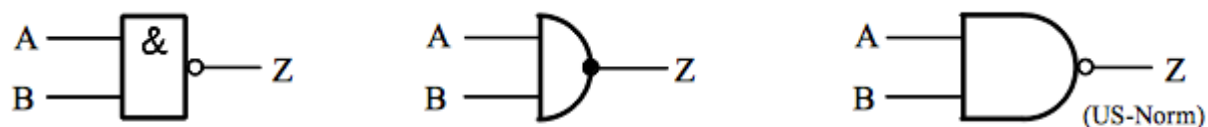
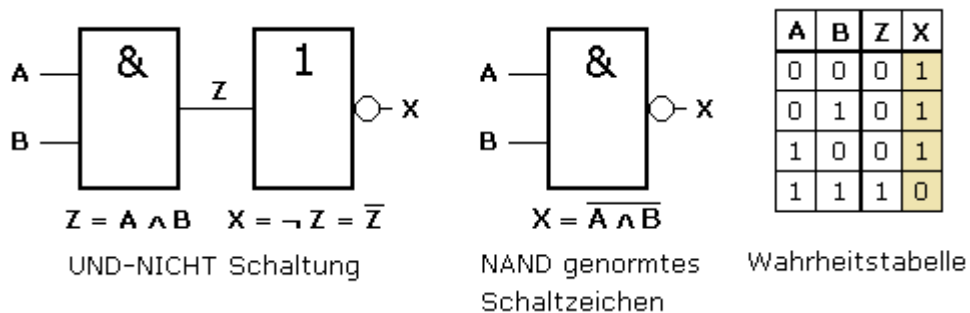


Bild 5-13: Schaltzeichen für NAND-Gatter

NOR-Verknüpfung

Eine Kombination aus einem **ODER-Gatter** mit nachfolgendem **NICHT-Gatter** ergibt ein **NOR-Gatter**. Die Ausgangsvariable Z des ODER-Gatters wird durch das NICHT-Gatter negiert und erzeugt die Ausgangsvariable X des NOR-Glieds. NOR-Glieder haben in logischen Schaltungen die gleiche wichtige Bedeutung wie NAND-Glieder.

Der Ausgangszustand eines NOR-Glieds ergibt 1, wenn alle Eingangszustände 0 sind.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)
NEGIERTES ODER	NOT OR	Negative Disjunktion	$Z = \overline{A \vee B}$

Schaltwerttabelle

a	b	$a \vee b$
0	0	1
0	1	0
1	0	0
1	1	0

Schaltfunktion

$$f(a,b) = a \vee b$$

Schaltsymbol

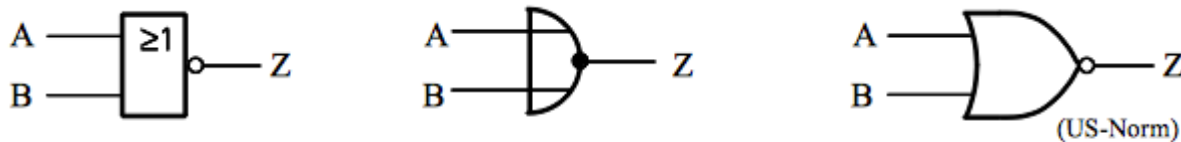
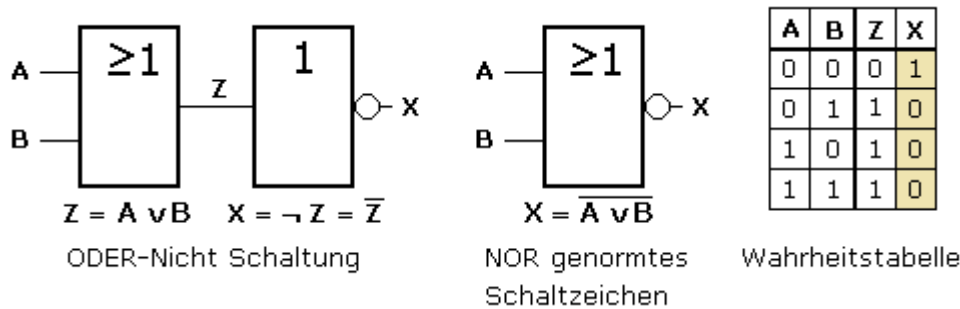


Bild 5-15: Schaltzeichen für NOR-Gatter

XOR-Verknüpfung

Die logische Verknüpfung des XOR-Gatters mit zwei Eingangsvariablen kann mit einem '**entweder - oder**' umschrieben werden. Die **Ausgangsvariable wird immer dann 'true' liefern, wenn die Eingangsvariablen unterschiedliche Zustände haben**. Die Wahrheitstabelle des XOR-Gatters entspricht dem ODER-Gatter mit dem Ausschluss gleicher Eingangszustände, also exklusiv der Äquivalenz. Dieses Verhalten wird als Antivalenz bezeichnet.

Der Ausgangszustand eines XOR-Glieds ergibt 1, wenn eine ungerade Zahl der Eingangszustände 1 und alle anderen Eingangszustände 0 sind.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)
Exklusives ODER (Antivalenz)	Exclusiv OR	Exklusive Disjunktion	$Z = a \vee b$

Schaltwerttabelle

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	0

Schaltfunktion

$$f(a,b) = a \vee b$$

Schaltsymbol

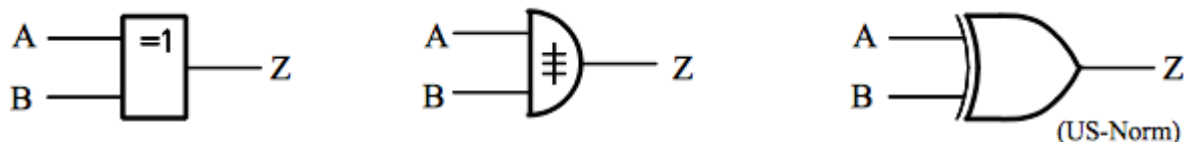
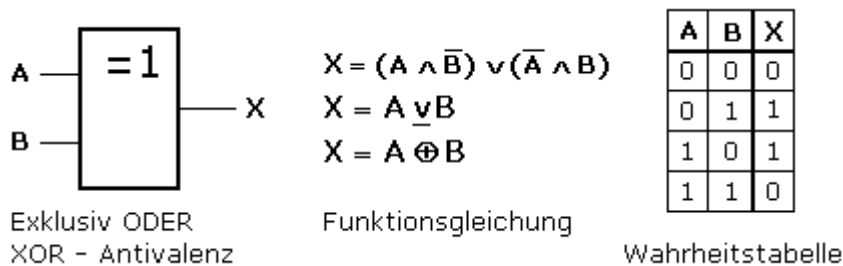


Bild 5-21: Schaltzeichen für ANTIVALENZ-Gatter

XNOR-Verknüpfung

Die Bezeichnung **Äquivalenz bedeutet Gleichwertigkeit**. Beim XNOR-Gatter mit zwei Eingangsvariablen ist der **Zustand der Ausgangsvariable 1, wenn beide Eingangsvariablen den gleichen Zustand 0 oder 1 haben**. Die Funktion kann durch eine Schaltung mit vier NOR-Gattern erreicht werden. Es sind zwei unterschiedliche Schaltzeichen zu finden.

Der Ausgangszustand eines XNOR-Glieds ergibt 1, wenn eine gerade Zahl der Eingangszustände 1 und alle anderen Eingangszustände 0 aufweisen oder alle 0 sind.

Zeichen

Deutsch	Englisch	Fachausdruck	Zeichen (boolsche Algebra)
Exklusives NICHT ODER (Äquivalenz)	Exclusiv NOT OR	Biimplikation	$Z = a \equiv b$

Schaltwerttabelle

a	b	$a \equiv b$
0	0	1
0	1	0
1	0	0
1	1	1

Schaltfunktion

$$f(a,b) = a \equiv b$$

Schaltsymbol

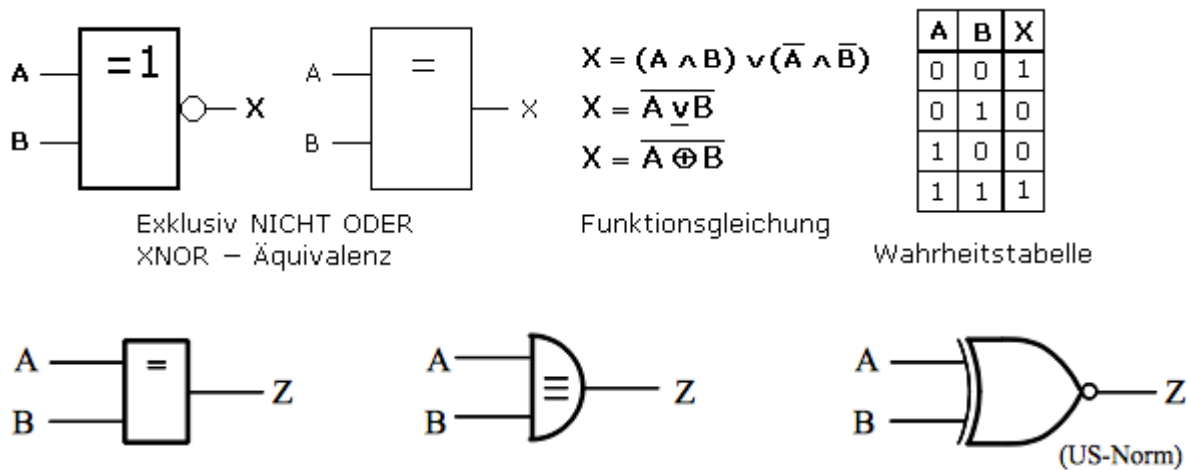


Bild 5-19: Schaltzeichen für ÄQUIVALENZ-Gatter

[xor_nand.swf](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_03



Last update: **2019/05/14 15:27**

Gesetze der Schaltalgebra

Vorrang- und Klammerregel

Wie in der Algebra ist auch in der Digitaltechnik auf eine bestimmte Reihenfolge der Operationen zu beachten. Die Negation sollte vor der Konjunktion (UND) und diese vor der Disjunktion (ODER) ausgeführt werden. Das ist vergleichbar mit der Aussage, dass die Multiplikation Vorrang vor der Addition hat.

Bei mehreren Variablen und unterschiedlichen Verknüpfungen kann eine Klammersetzung notwendig sein. Beachtet man die Vorrangregel, kann man auf viele Klammern verzichten. Ein Setzen zeigt sogleich eindeutig, welche der Variablen wie zu verknüpfen ist. Bei ODER sollte immer geklammert werden, ebenfalls beim Anwenden der De Morganschen Gesetze auf NAND- und NOR-Verknüpfungen.

$X = A \vee B \wedge C$	A	B	C	$B \wedge C$	$A \vee (B \wedge C)$	$A \vee B$	$(A \vee B) \wedge C$
nach der Vorrangregel gilt	0	0	0	0	0	0	0
$X = A \vee (B \wedge C)$	0	0	1	0	0	0	0
	0	1	0	0	0	1	0
	0	1	1	1	1	1	1
festgelegte Abfolge der Verknüpfungen	1	0	0	0	1	1	0
$Z = (A \vee B) \wedge C$	1	0	1	0	1	1	1
	1	1	0	0	1	1	0
	1	1	1	1	1	1	1

$A \vee (B \wedge C) \neq (A \vee B) \wedge C$

Bei drei Eingangsvariablen und binären Zuständen gibt es $2^3 = 8$ Eingangskombinationen. Die Wahrheitstabelle zeigt bei definierter Klammersetzung oder Beachtung der Vorrangregel UND vor ODER unterschiedliche Ergebnisse.

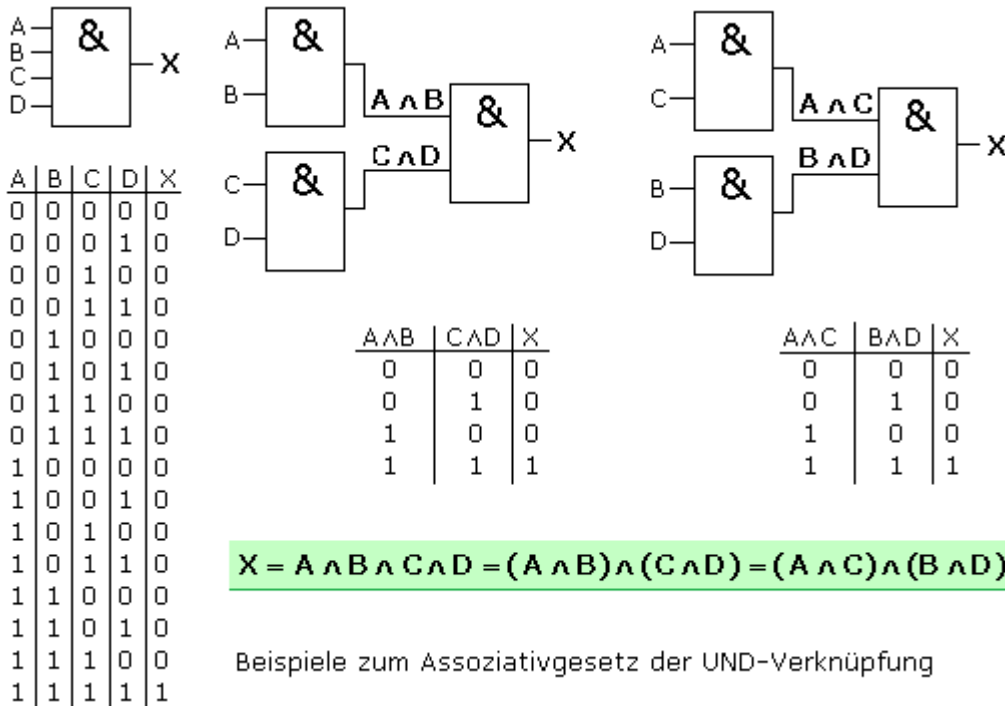
Kommutativgesetz

Das Kommutativ- oder Vertauschungsgesetz besagt, dass bei der UND sowie ODER Verknüpfung auch bei beliebiger Vertauschung der Reihenfolge der Eingangsvariablen das Ergebnis gleich bleibt.

UND	$X = A \wedge B \wedge C = C \wedge A \wedge B = B \wedge C \wedge A$
ODER	$X = A \vee B \vee C = C \vee A \vee B = B \vee C \vee A$

Assoziativgesetz

Das Assoziativ- oder Verbindungsgesetz besagt, dass bei einer UND- beziehungsweise ODER-Verknüpfung mit mehr als zwei Schaltvariablen die Verknüpfung auch stufenweise nacheinander in beliebiger Reihenfolge erfolgen kann.



Anstelle des UND-Gatters mit vier Eingangsvariablen lassen sich zwei UND-Gatter mit zwei Eingängen verwenden. Die Variablen A, B, C, D werden einzeln beliebig mit den Eingängen verbunden. Die Ausgangsvariable der beiden UND-Gatter kann den Wert 0 oder 1 annehmen. Beide Ausgangsvariablen sind nochmals mit einem UND-Gatter zu verknüpfen, wobei sich wieder vier Eingangskombinationen ergeben. Nur wenn alle Eingangsvariablen 1 sind, wird der Ausgangszustand ebenfalls 1.

Das Assoziativgesetz gilt gleichermaßen für die ODER-Verknüpfung. Die Kammersetzung ist nicht notwendig und soll nur verschiedene Verteilungen besser erkennbar machen.

Distributivgesetz

Das Distributiv- oder Verteilungsgesetz wird zur Vereinfachung von Verknüpfungsgleichung angewendet. Es ist vergleichbar mit dem Ausmultiplizieren und Ausklammern von Variablen der normalen Algebra. Da die logische UND-Verknüpfung der algebraischen Multiplikation entspricht, während die ODER-Verknüpfung mit der Addition vergleichbar ist, gibt es zwei unterschiedliche Distributivgesetze.

Konjunktives Distributivgesetz

Eine Variable wird mit UND verknüpft und auf den Folgeausdruck verteilt. Die Vorgehensweise entspricht dem aus der Algebra bekannten Ausmultiplizieren eines Klammersausdrucks mit einem Faktor. Im umgekehrten Fall kann eine Variable, die mit mehreren anderen Variablen verknüpft ist, ausgeklammert werden. Das bedeutet für den Schaltungsaufbau eine Einsparung an Gattern.

$$X = A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$$

→ Ausmultiplizieren
← Ausklammern

$$X = A \wedge (A \vee B)$$

$$X = (A \wedge A) \vee (A \wedge B)$$

$$X = A \vee (A \wedge B) = A$$

$$X = A$$

UND		ODER		
A	B	A ∧ B	A ∨ B	X
0	0	0	0	0
0	1	0	1	0
1	0	0	1	1
1	1	1	1	1

Beim Ausklammern wird die Variable mit ihrem Verknüpfungszeichen, hier UND, vor die Klammer gesetzt. Die in der Klammer stehenden Variablen werden mit dem zuvor zwischen den Klammern stehenden ODER verknüpft.

Disjunktives Distributivgesetz

Eine Variable kann durch ein vergleichbares Ausmultiplizieren auf andere Ausdrücke verteilt werden oder bei mehrfachem Auftreten ausgeklammert werden. Beim Ausklammern wird das ODER Verknüpfungszeichen mit der Variablen vor die Klammer geschrieben. disjunktives Distributivgesetz

$$X = A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$$

→ Ausmultiplizieren
← Ausklammern

$$X = A \vee (A \wedge B)$$

$$X = (A \vee A) \wedge (A \vee B)$$

$$X = A \wedge (A \vee B)$$

$$X = A$$

ODER		UND		
A	B	A ∨ B	A ∧ B	X
0	0	0	0	0
0	1	1	0	0
1	0	1	0	1
1	1	1	1	1

De Morgansche Gesetze

Die boolesche Algebra wurde vom englischen Mathematiker De Morgan weiter entwickelt. Für die Schaltalgebra gibt es zwei De Morgansche Gesetze. Sie machen Aussagen zur Negation einer Verknüpfung und der Umkehr von Verknüpfungszeichen. Mit den De Morganschen Gesetzen lassen sich bei der Entwicklung von Digitalschaltungen Gatter gegeneinander austauschen, Schaltungen verkleinern oder mit nur einem Gattertyp verwirklichen. Das erste Gesetz ist für die NAND-Verknüpfung und das zweite Gesetz entsprechend für die NOR-Verknüpfung definiert.

$$X = \overline{A \wedge B} = \overline{A} \vee \overline{B}$$

1. De Morgansche Gesetz

ODER		UND		
A	B	A ∨ B	A ∧ B	X
0	0	0	0	1
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

$$X = \overline{A \vee B} = \overline{A} \wedge \overline{B}$$

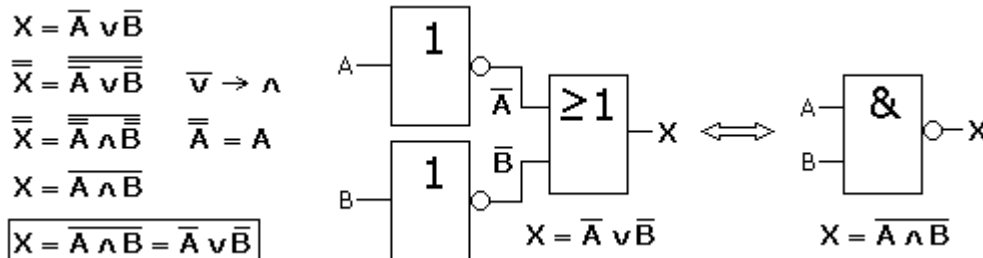
2. De Morgansche Gesetz

UND		ODER		
A	B	A ∧ B	A ∨ B	X
0	0	0	0	1
0	1	0	1	0
1	0	0	1	0
1	1	1	1	0

Ist eine Verknüpfung insgesamt negiert, so ist ihre Ausgangsvariable negiert. Die Negation kann auf die Einzelglieder aufgeteilt werden, wobei aus der Grundverknüpfung UND ein ODER beziehungsweise aus ODER ein UND wird. Eine doppelte Negation hebt sich auf. Getrennte Negationsstriche über

Variablen bedeuten, dass die Eingangsvariablen negiert sind.

Die oben zum 1. De Morganschen Gesetz gezeigte Verknüpfung kann schaltungstechnisch mit zwei NICHT- und einem ODER-Gatter verwirklicht werden. Wie zu erkennen ist, folgt das gleiche Ergebnis mit nur einem NAND-Gatter. Für das 2. De Morgansche Gesetz gilt die entsprechende Aussage. Zwei NICHT- und ein UND-Gatter reduzieren sich auf den Einsatz eines NOR-Gatters. Mit der doppelten Negation und den De Morganschen Gesetzen kann bei einer gegebenen Funktionsgleichung auf das Bestehen einer derartige Vereinfachung geprüft werden. Treten dabei mehrfache Negationen auf, so lassen sie sich von innen nach außen auflösen.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_04



Last update: 2019/05/14 15:28

Digitale Rechenschaltungen

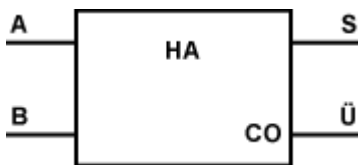
Aus logischen Verknüpfungen lassen sich digitale Schaltungen zusammenbauen, mit denen man Rechengänge durchführen kann. Das heißt, diese Schaltungen haben zwischen ihren Eingängen eine Kombination aus logischen Verknüpfungen, die einem Rechengang entspricht. In der Digitaltechnik kennt man Rechenschaltungen hauptsächlich für das duale Zahlensystem und den BCD-Code. Im Prinzip kann für jedes Zahlensystem eine Rechenschaltung aufgebaut werden.

Einige Rechenschaltungen der Digitaltechnik

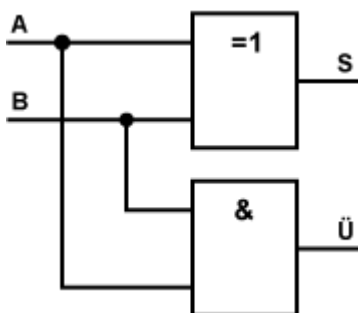
- Addiererschaltungen
- Subtrahiererschaltungen
- Addier-Subtrahier-Werke
- Multiplikationsschaltungen
- Arithmetisch-logische Einheit (ALU)

Stellvertretend für alle Rechenschaltungen dienen die folgenden Ausführungen zum Halbaddierer und dem Volladdierer.

Halbaddierer



Ein Halbaddierer ist die einfachste Rechenschaltung und kann zwei einstellige Dualziffern addieren. Der Eingang A des Halbaddierers ist der Summand A, der Eingang B ist der Summand B. Die Schaltung hat zwei Ausgänge. Der Ausgang S als Summenausgang (2^0) und der Ausgang Ü als Übertrag (2^1) für die nächsthöhere Stelle im dualen Zahlensystem.



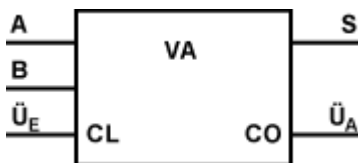
B	A	Ü	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Der Halbaddierer ist eine Schaltung aus den Verknüpfungsgliedern XOR und UND. Das XOR ist die Addier-Verknüpfung. Das UND stellt fest, ob ein Übertrag für die nächsthöhere Stelle vorgenommen

werden muss. Aus der Tabelle ist ersichtlich, dass das Ergebnis aus der Spalte Summe (S) einer Exklusiv-ODER-Verknüpfung (Antivalenz, XOR) entspricht. Das Ergebnis der Spalte Übertrag (Ü) entspricht einer UND-Verknüpfung. Die so entstandene Schaltung wird als Halbaddierer bezeichnet. Sie ist in der Lage zwei 1-Bit Summanden (einstellig) zu addieren.

Volladdierer

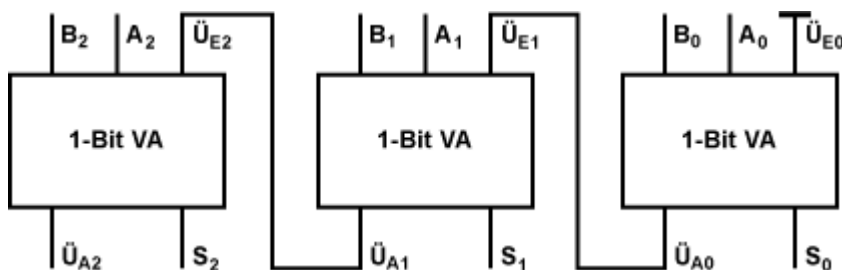
Um **mehrstellige Dualzahlen** addieren zu können benötigt man Schaltungen die auch einen Übertrag einer niederwertigen Stelle berücksichtigen. Man spricht vom **Übertragseingang ÜE**. Die Schaltung bezeichnet man als **Volladdierer (VA)**. Ein Volladdierer kann drei Dualzahlen addieren. Bzw. zwei Dualzahlen addieren und den Übertrag aus einer niederwertigen Stelle berücksichtigen.



Folgende Wahrheitstabelle ergibt sich:

ÜE	B	A	ÜA	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

3-Bit-Volladdierer



Folgende Schaltung zeigt einen 3-Bit-Volladdierer (VA), der aus drei 1-Bit-Volladdierer (VA) realisiert wurde. Damit lassen sich zwei dreistellige Dualzahlen addieren. Der Eingang \bar{U}_{E0} liegt an 0 V (Masse), weil in der niederwertigsten Stelle kein Übertrag berücksichtigt werden muss.

Beispiel

010 = A
+111 = B

$$1001 = S$$

0. Stelle

\ddot{U}_{E0}	A_0	B_0	\ddot{U}_{A0}	S_0
0	0	1	0	1

1. Stelle

\ddot{U}_{E1}	A_1	B_1	\ddot{U}_{A1}	S_1
0	1	1	1	0

2. Stelle

\ddot{U}_{E2}	A_2	B_2	\ddot{U}_{A2}	S_2
1	0	1	1	0

Ergebnis
 $\ddot{U}_{A2} S_2 S_1 S_0$

1 0 0 1

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_05
Last update: **2019/05/14 15:28**

2.06) Übungen

1) Um ihren Swimmingpool schneller füllen zu können, hat Fam. Huber zwei Wasserleitungen (beide verfügen über einen Absperrhahn B und C verlegt, die direkt in den Pool führen. Diese Leitungen zweigen beide von einer Hauptleitung ab, die ebenfalls über einen Absperrhahn A verfügt. Demnach kann Wasser nur in den Pool fließen, wenn der Hauptabsperrhahn A und zumindest einer der anderen Absperrhähne offen sind.

Stelle die Situation mittels einer Wahrheitstabelle und einer Schaltfunktion (LogikSim, ...) dar.

2) Frau Müller sagt zu ihrem Mann: „Den ganzen Sonntag hockst du auf dem Sofa, trinkst Bier und stopfst dir Kartoffelchips in den Schlund! Du solltest mal joggen!

Ihr Gatte erwidert: „Ich verspreche dir für nächsten Sonntag Folgendes:

- Wenn ich jogge, werde ich sogar auf Bier oder auf Chips verzichten.
- Wenn ich keine Chips esse, trinke ich kein Bier oder ich jogge nicht.
- Wenn ich aber jogge, brauche ich unbedingt hinterher ein Bier!
- Allerdings: Auch falls ich nicht jogge, werde ich auf jeden Fall Bier oder Chips zu mir nehmen.“

Falls Herr Müller seine Versprechen einhält, wie könnten seine Aktivitäten am nächsten Sonntag aussehen? Erstelle einen schaltalgebraischen Ausdruck und die Wahrheitstabelle zu diesen Aussagen.

3) Zeichne für folgende schaltalgebraischen Ausdrücke die zugehörige Schaltfunktion bzw. erstelle die jeweilige Wahrheitstabelle!

- $f(A,B) = (\neg A \vee \neg B)$
- $f(A,B,C) = (\neg A \wedge \neg B) \wedge C$
- $f(A,B) = (\neg A \vee \neg B) \vee (\neg A \wedge B)$

4) Überprüfe das Gesetz von De Morgan mit Hilfe einer Leitwerttabelle und stelle die Schaltung im Digitalsimulator dar.

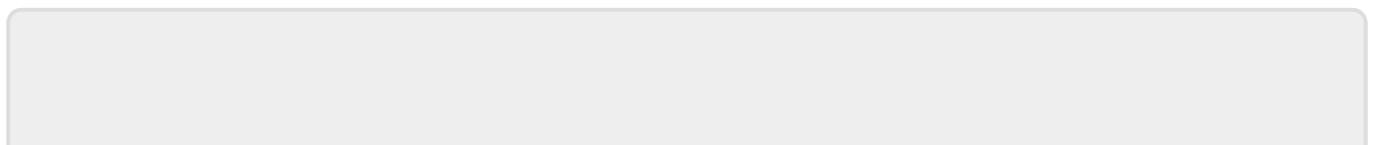
5) Überprüfe das Distributivgesetz mittels einer Leitwerttabelle und stelle die Schaltung im Digitalsimulator dar.

6) Vereinfache die Schaltfunktion $f(a,b) = a \wedge (\neg a \vee b)$ und baue die Schaltung im Digitalsimulator auf.

7) Vereinfache die Schaltfunktion $f(a,b) = (a \vee b) \wedge (a \vee \neg b) \wedge (\neg a \vee b)$

8) Vereinfache die Schaltfunktion $f(a,b) = (a \wedge b) \vee (a \wedge \neg b) \vee (\neg a \wedge b)$

9) Zur automatischen Brandbekämpfung wurden drei Sensoren in einem Raum angebracht. Melden mindestens zwei der drei Sensoren eine Rauchentwicklung, so schaltet sich die Sprinkleranlage ein. Entwirf eine Leitwerttabelle, eine Schaltfunktion und stelle diese mittels Digitalsimulator dar!



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:02:2_07



Last update: **2019/05/14 15:28**

3) Audiotbearbeitung

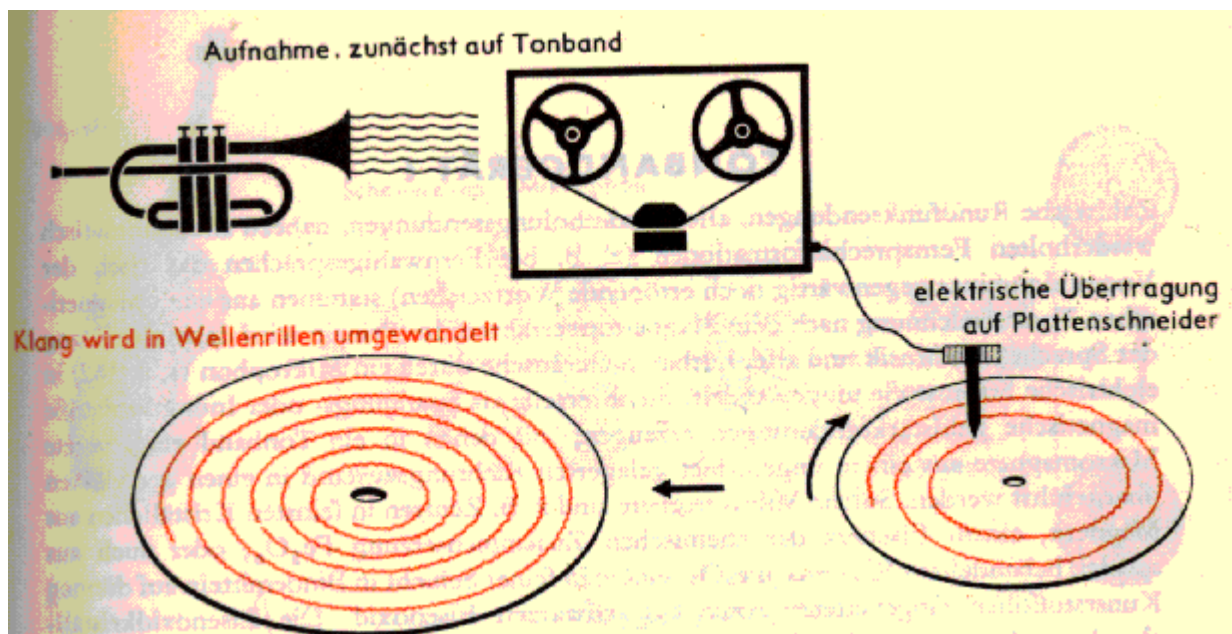
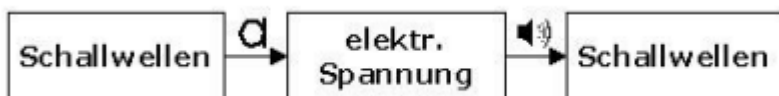
3.1) Theorie

Schallwellen

Schallwellen sind Druckschwankungen in einem Übertragungsmedium (z.B. Luft, Wasser), die von uns wahrgenommen werden können.

In der Analogtechnik wird der Schalldruck durch eine dazu proportionale elektrische Spannung repräsentiert.

- ein Mikrofon wandelt den Schalldruck in die elektrische Spannung um
- diese Spannung kann in andere analoge Darstellungsformen überführt werden
 - Magnetisierung (auf Tonband)
 - Ausleuchtung (auf die Tonspur beim Film)
 - mechanische Auslenkungen (zur Erstellung einer Schallplatte)
- am Ende übersetzt ein Lautsprecher die Spannung wieder in einen dazu analogen Wecheldruck



Ton

Ein Ton ist eine Schwingung, die durch periodische Regelmäßigkeit eine definierbare Grundfrequenz - die Tonhöhe - hörbar werden lässt. Der Klang eines Tons hängt von den Anteilen weiterer Schwingungen innerhalb der Grundschwingung ab. Ein Geräusch - etwa einer Rassel - hat dagegen keine periodische Grundschwingung, die Luftdruckschwankungen verlaufen so unregelmäßig, dass

keine Tonhöhe definierbar ist. Die Übergänge zwischen Ton und Geräusch sind fließend, denkt man daran, dass sich Konzertpauken sehr wohl stimmen lassen, obwohl sie als perkussive Instrumente bereits sehr diffuse Klangspektren erzeugen.

Sound

Sound wird heute oft im 'Multimedia-Deutsch' für alles, was mit dem Audibereich zu tun hat verwendet. Zum Teil geschieht das sicher aus Unkenntnis der Begriffe aus Akustik und Musik. Der Begriff **Sound** hat jedoch auch Vorteile, da er problematische Fragen der traditionellen Begriffe und Wissenschaftsdisziplinen ausblendet (Was ist Musik?!).

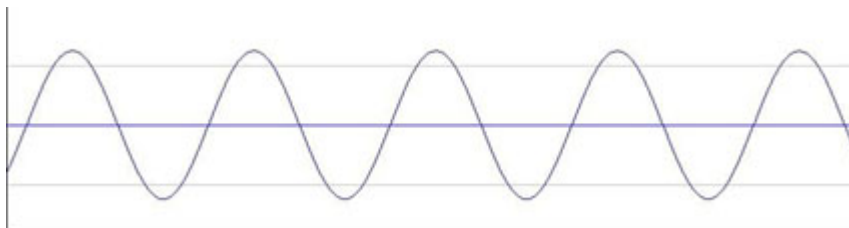
Sprache

Sprache ist danach ein geräuschhaftes=nichtperiodisches Klangereignis, das durch seine definierten Frequenzgemische sprachliche Lautmuster enthält. Im Gesang werden diese Lautmuster einer periodischen Schwingung aufmoduliert. Für gute Sprachverständlichkeit reicht ein Frequenzbereich bis 8 kHz vollkommen aus.

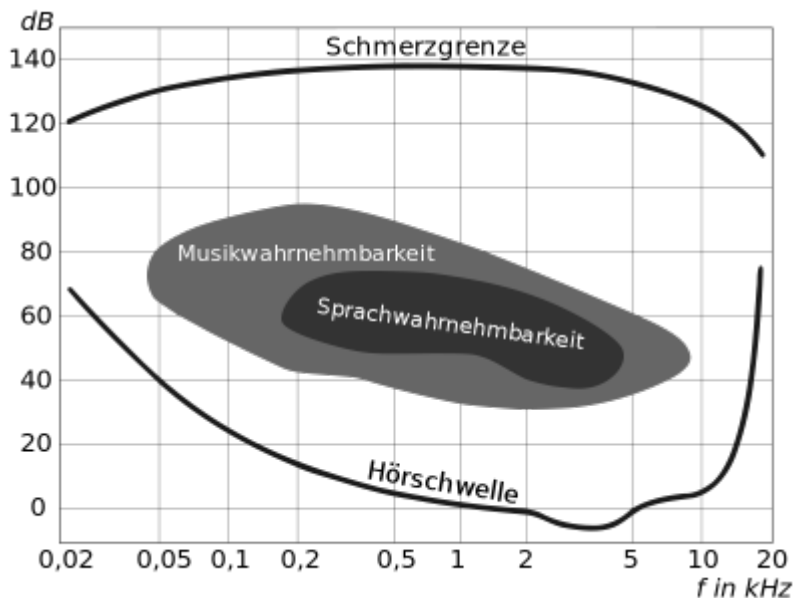
Beispiele für Schwingungen im Audibereich

Sinuswelle

Das Urbild einer periodischen Schwingung, sie besitzt keine weiteren Frequenzanteile.



Hörbereich



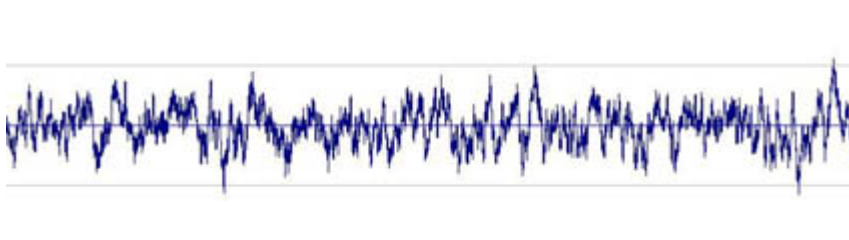
Sinuswellenton in verschiedenen Frequenzen → Teste dein Gehör!



Sinuswelle in verschiedenen Frequenzen (20Hz bis 20kHz)

Rauschen

Schwingungschaos = das Gegenteil einer periodischen Schwingung.



Digitalisierung von Audio

Grundsätzlich sind Daten im Audibereich zeitkritisch, d.h. sie stellen Anforderungen an Hard- und Software, die historisch gesehen atypisch für einen 'Rechner' sind. Bis heute sind damit

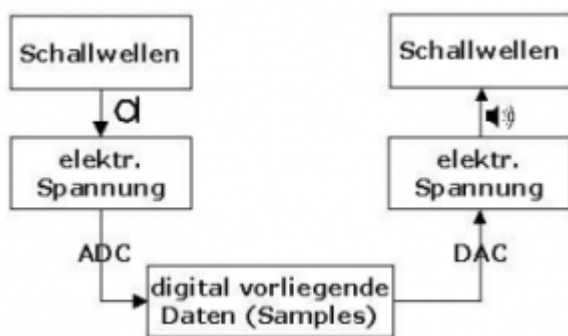
Geschwindigkeits-, Bandbreiten- und Speicherprobleme verbunden, die von technischen und ökonomischen Rahmenbedingungen abhängen.

Um den Sound am Computer bearbeiten zu können, müssen die analogen elektrischen Signale in digitale Daten umgewandelt werden.

Damit Audiodaten aufgenommen und wiedergegeben werden können, werden **A(nalog)/D(igital)-Wandler (AD-Converter)** bzw. **D/A-Wandler (DA-Converter)** benötigt. Diese werden i.d.Regel von der Soundkarte zur Verfügung gestellt.

- hierbei wird die analoge Spannung in einem regelmäßigen Zeittakt (**Samplingfrequenz**) gemessen.
- Diese Messwerte sind je nach ausgewählter **Bitrate** aufgelöst
- heraus kommt eine zahlencodierte Nachbildung des analogen Signals als eine **Abfolge von diskreten Schritten**

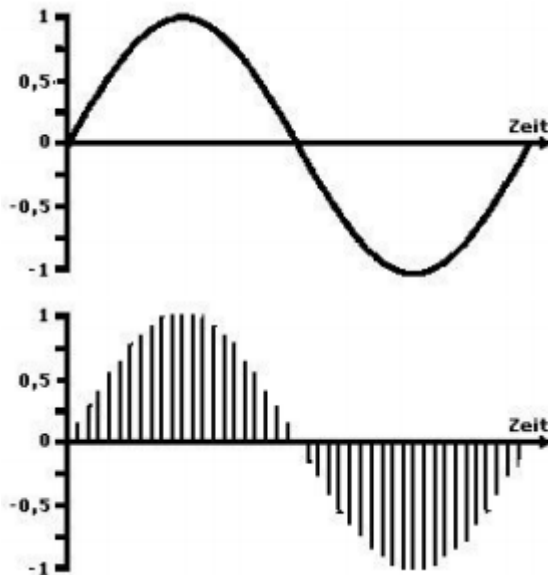
Beim Abspielen dieser Meßwerte wandelt ein Digital-AnalogConverter (DAC) den Zahlencode wieder in eine Spannung um, die vom Lautsprecher wieder in Schalldruck umgesetzt wird.



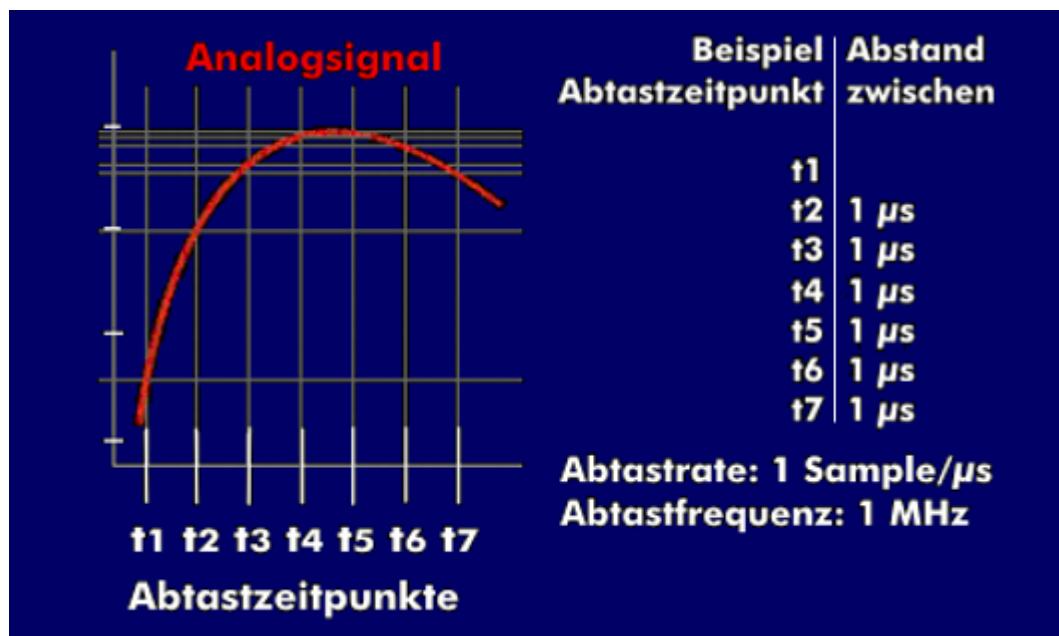
Beim Computer befinden sich die beiden Converter in der Soundkarte oder sind schon direkt im Mainboard integriert.

Samplingrate/Samplingfrequenz

Bei der Digitalisierung werden in **regelmäßigen zeitlichen Abständen Messproben (Samples)** des analogen Ausgangssignals entnommen. Die **Abtastrate** wird **pro Sekunde** angegeben.



z.B. :Für die Audio-CD wird mit 44.100 Hz (also 44100 mal in der Sekunde) gemessen.



Aliasing

Beim Digitalisieren können Frequenzen entstehen, die im Original nicht vorhanden sind. Diese Artefakte oder 'Aliasfrequenzen' des Digitalisierungsvorgangs sind Ergebnisse einer **unzureichenden Abtastfrequenz**. Die **Abtastrate** muss **mindestens das Doppelte der zu digitalisierenden Frequenz** (siehe Abtasttheorem nach Nyquist/Shannon) betragen. Je höher die Abtastrate, desto besser ist die Audioqualität nach der Digitalisierung.

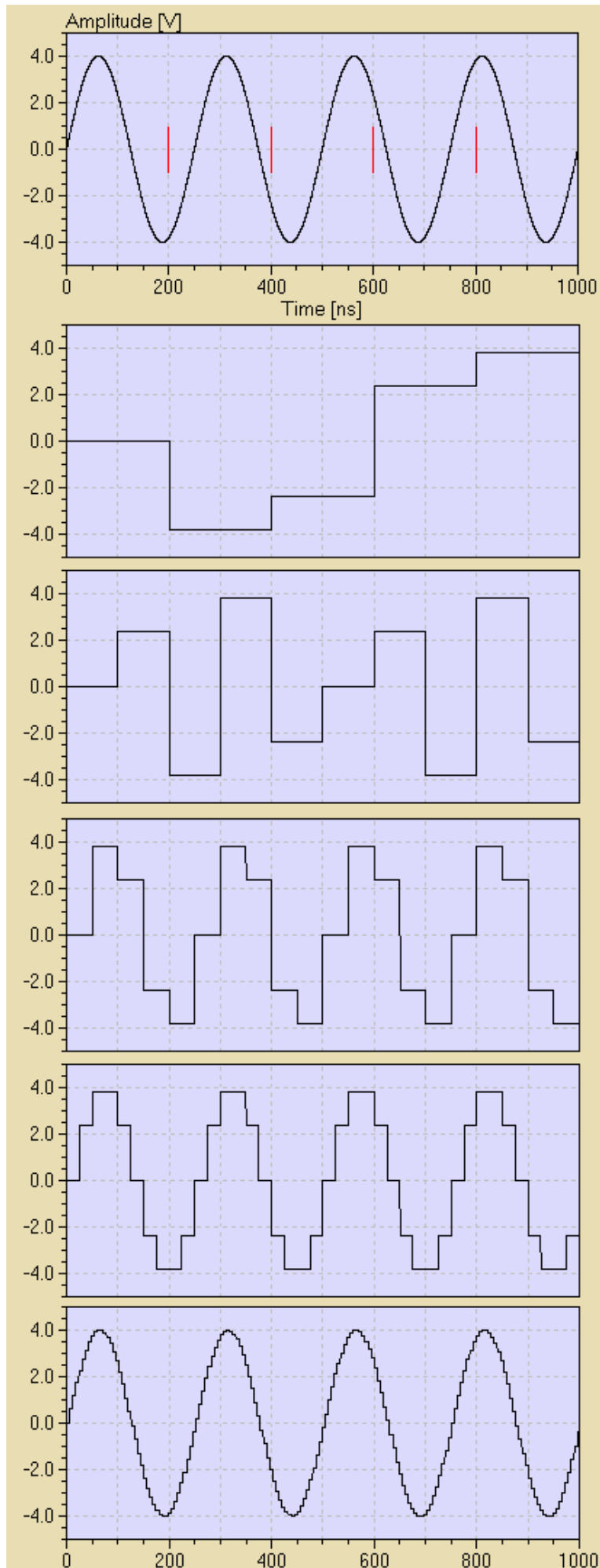
Beispiel

Das folgende Originalsignal zeigt in der Skizze eine Sinuswelle mit der Schwingungsdauer von $T=250\text{ns}$.

Das heißt die Frequenz ($1/s = \text{Hz}$) des Originalsignals ist:

$$1/250\text{ns} \Rightarrow 1/250 \cdot 10^{-9} \text{ s} \Rightarrow 1/250 \cdot 10^{-9} \text{ s} \Rightarrow 4000000 \cdot 1/\text{s} \Rightarrow 4000000 \text{ Hz} \Rightarrow 4 \text{ MHz}$$

D.h. Um ein 4-MHz-Signal korrekt wiedergeben zu können ist eine minimale Abtastrate von 8 MHz notwendig. Wird diese Grenzfrequenz unterschritten, entstehen Artefakte, die mit dem ursprünglichen Signal nichts mehr zu tun haben.



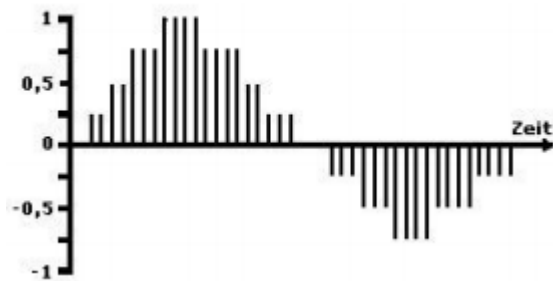
Bittiefe

Für die jeweiligen Samples stehen nur eine begrenzte Anzahl an Messwerten zur Verfügung. Die für das Digitalisieren gewählte Bittiefe bestimmt diese Menge, die sich über den Messbereich verteilt.

Mit der **Bittiefe ist die Länge der Binärzahlen** gemeint, aus denen die digitale Audiodatei letztendlich besteht. Binärzahlen sind Zahlen, die nur aus Nullen und Einsen bestehen. Ein Computer rechnet in seinem Inneren nur mit diesen Zahlen.

- 1 Bit $\rightarrow 2^1 = 2$ Zustände (0 oder 1)
- 2 Bit $\rightarrow 2^2 = 4$ Zustände (0 bis 3)
- 3 Bit $\rightarrow 2^3 = 8$ Zustände (0 bis 7)
- ...

Abtastung mit Bittiefe = 3



unkomprimierte Audioformate

Für unkomprimierte Audiosignale gibt es für den Computer die Formate **WAV(Windows)**, **AIFF (Macintosh)**, **AU (Linux)** und **PCM**.

Diese Formate unterscheiden sich nicht wesentlich. Sie können Audio prinzipiell in jeder Samplingfrequenz und Bittiefe speichern. Die Wahl richtet sich dabei nach der gewünschten Wiedergabequalität, der Downloadgeschwindigkeit und der Speicherplatzverfügbarkeit.

Üblich sind folgende Kombinationen:

Qualität	Bittiefe	Samplingfrequenz Hz	Anzahl Kanäle	Downloadrate kbit/s	Speicherbedarf MByte/min
Telefon	8	8000	1 (Mono)	62,5	0,5
Radio	16	22050	1 (Mono)	352,8	2,8
CD	16	44100	2 (Stereo)	1378,1	11
DVD	24	48000	6 (5.1-Sound)	6750,0	54

komprimierte Audioformate

Für die meisten Anwendungen ist der Speicherverbrauch unkomprimierter Formate zu hoch (z.B. für Downloads aus dem Internet, als Soundtrack für Videos auf CD oder DVD)

Für solche Fälle gibt es Formate, welche die Datenmenge reduzieren, indem sie nicht hörbare Frequenzen eliminieren.

Diese Verfahren verschlechtern die Qualität mit jedem Kompressions- und Dekompressionsvorgang und sind daher für die Zwischenspeicherung nicht zu empfehlen. Es sollte daher beim Editieren möglichst lange mit der unkomprimierten Datei gearbeitet werden, bis man das Endresultat schließlich für die finale Anwendung passend komprimiert.

Beispiele für die Anwendung vom Komprimierungs-Codecs:

Anwendung	Qualität	Codecs	Original kbit/s	Komprimiert kbit/s	Kompressionsfaktor
Internet Echtzeit-Streaming	Telefon bis Radio	mp3, mpeg4, wma, RealAudio, ogg, vorbis	689-1378	30-60	ca. 11-45
Internet Download	quasi-CD	mp3, wma, ogg, vorbis	1378	128	10,7
DVD	quasi-DVD	AC3, mpeg2	6750	448 (AC3)	15(AC3)

Streamingdienste

Streaming Audio ist eine Variante des Streaming Media, bei der Audiodaten kontinuierlich über ein Rechnernetz übertragen werden. In seiner einfachsten Form kann man sich Streaming Audio als Hörfunk im Internet vorstellen.

Musik-Streaming beschreibt die Option, jederzeit stationär oder mobil auf viele Millionen Musikstücke zugreifen zu können. Die Songs und Alben lagern in der Cloud und können via Internetverbindung bei verschiedenen Anbietern auf Anfrage abgerufen werden. Das Prinzip des Musikhörens auf Abfrage per Flatrate liegt voll im Trend. Eindeutige Indizien liefern stetig steigende Nutzerzahlen der Abo-Dienste. Die wichtigsten Anbieter der Branche verzeichnen ein konstantes bis atemberaubendes Wachstum. Ein Ende der Erfolgsgeschichte ist vorerst nicht abzusehen.

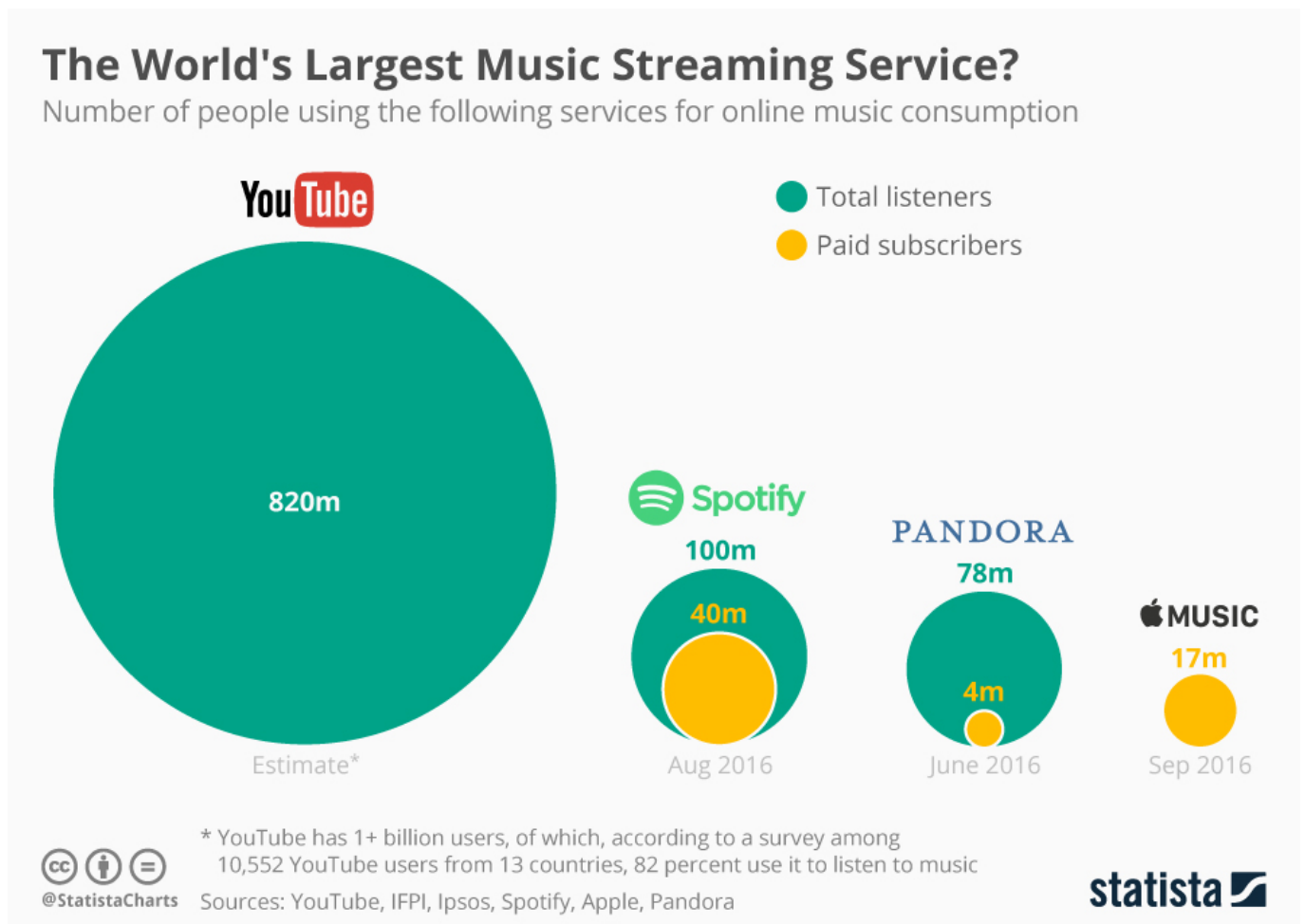
Spotify nennt diesbezüglich mittlerweile Zahlen weit jenseits der 100 Millionen-Marke. Auch Apple Music ist auf dem Vormarsch und spricht von knapp 30 Millionen zahlenden Nutzern. Immer mehr Menschen setzen auf Musik-Streaming und lassen die CD oder auch Schallplatte im Regal verstauben. Doch die Konkurrenz schläft nicht. Der Markt hält zahlreiche weitere Angebote bereit. So buhlen unter anderem Google mit Play Music, Amazon mit Prime Music und Music Unlimited, Napster, Deezer, Tidal oder auch SoundCloud Go um die Gunst der Hörerschaft.

Problemfall: Youtube

Nachdem der digitale Musikkonsum nun den Großteil des Marktes ausmacht, versucht die Branche, hier die Einnahmen zu steigern. Die Musikindustrie beklagt hier vor allem einen sogenannten Value Gap.

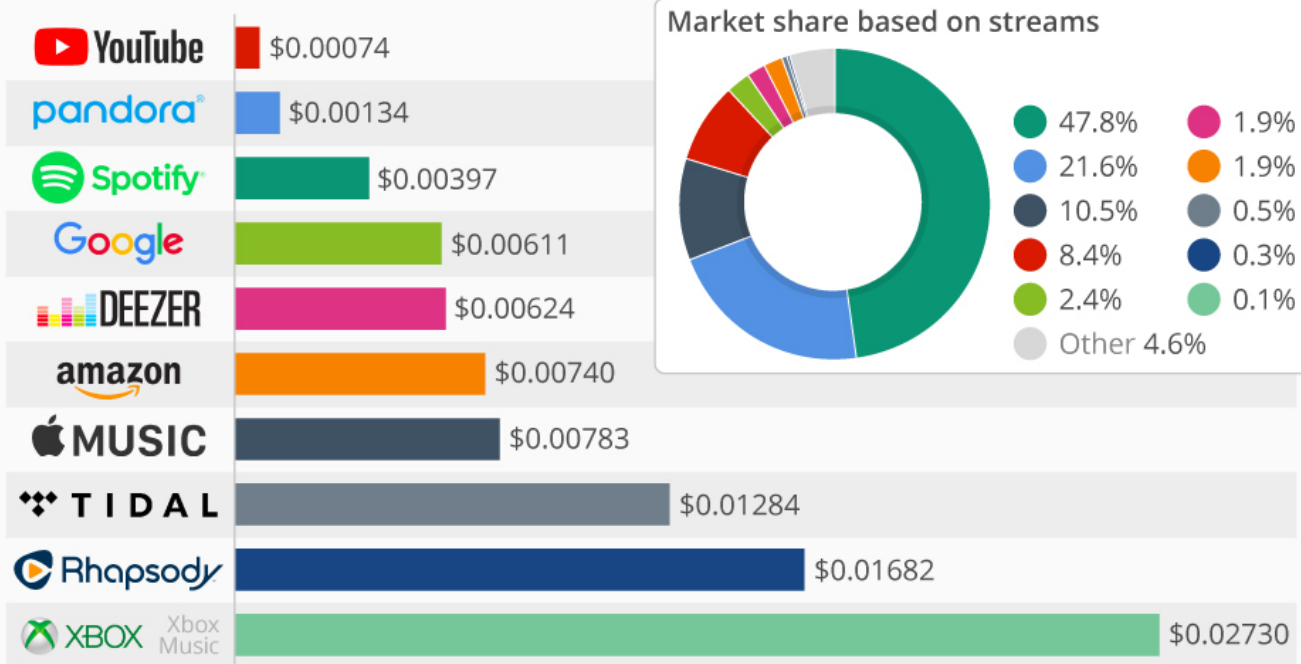
Wie die Industrie in ihrem globalen Jahresreport darlegt, würde zwar ein Großteil der gestreamten Musik über Googles Videoportal YouTube konsumiert. Von dort käme aber nur ein winziger Teil der Einnahmen. „Es muss endlich geregelt werden, dass sie am Markt Lizenzen erwerben müssen“, sagt

BVMI-Chef Florian Drücke.



Music-Streaming: Who Pays Best?

Gross payout to labels per stream by selected streaming services*



* Data based on a mid-sized indie label with an approximately 200+ album catalog now generating over 200m+ streams annually.
 @StatistaCharts Source: The Trichordist

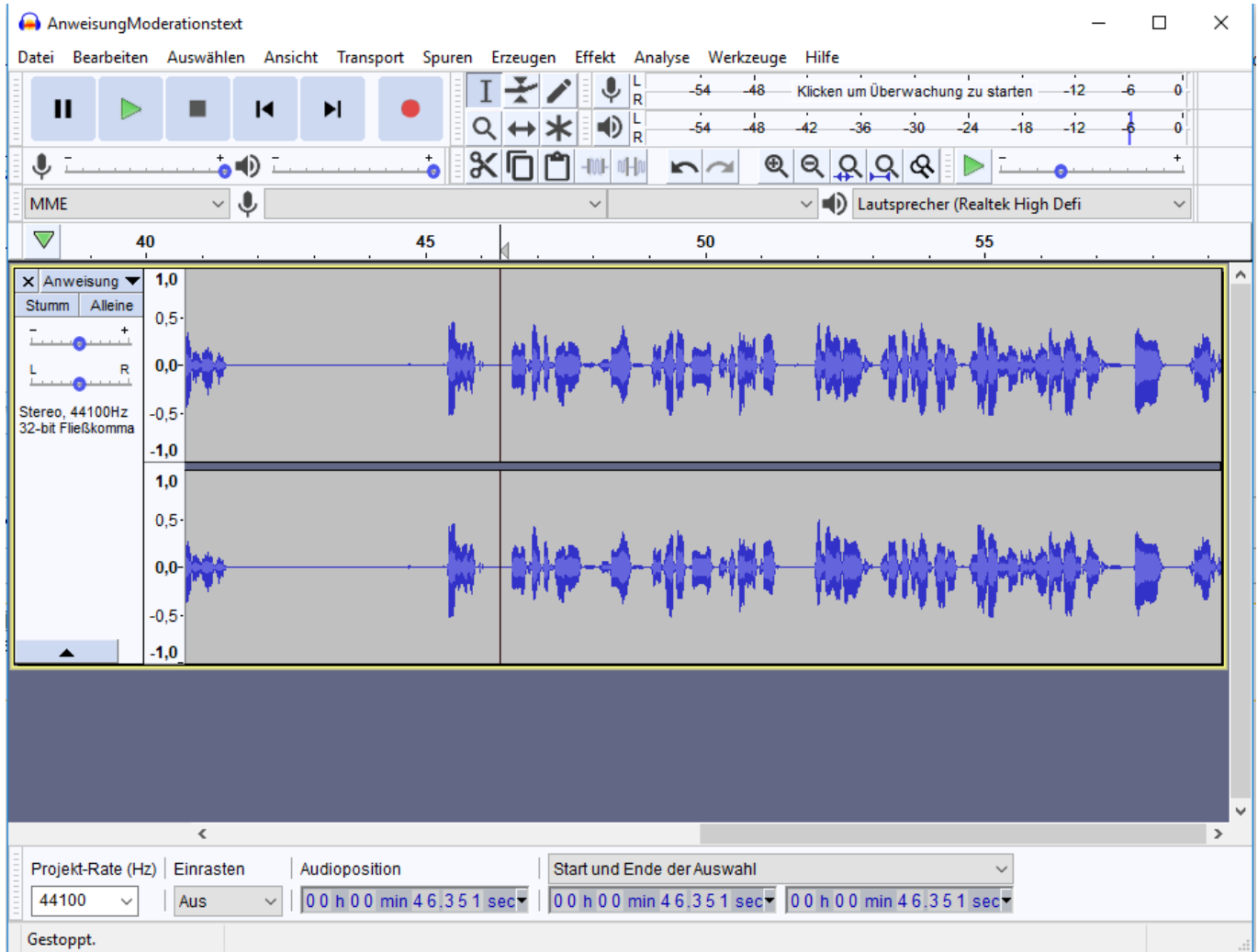
statista

3.2) Praxis mit Audacity

Die Audiotbearbeitung gehört zu den Grundlagen der Musikproduktion und wird auch »Audio Editing« genannt. Die Bearbeitung von Audiodateien findet in der Regel vor dem Abmischen statt. Allerdings spricht man auch von Audiotbearbeitung, wenn man nur Knackser aus MP3 Dateien entfernen oder MP3s schneiden möchte.

Ein mögliches Programm für Audio Editing ist:

- Programm Audacity
- Eine Anleitung auf YouTube
- Eine Reihe kurzer Filme zur Einführung in Audacity -> mit Microsoft Edge öffnen
- Shortcuts Audacity
- Übungen



Destruktive vs. Non-destruktive Audiotbearbeitung

Sehr wichtig für deine Audiodateien ist die Unterscheidung zwischen **destruktiver** und **non-destruktiver Audiotbearbeitung**. Als **destruktiv** wird die Bearbeitung von Audiodateien dann bezeichnet, wenn die **Änderung an der Originaldatei** selbst durchgeführt und festgeschrieben werden (Bsp. Audacity). Ist die Änderung erst einmal gemacht, lässt sie sich **nicht mehr rückgängig** machen.

Bei der **non-destruktiven Audiotbearbeitung** werden die Änderungen **nicht fest in die Datei geschrieben** (Bsp. Reaper). Das Programm zur Bearbeitung merkt sich in diesem Fall nur, wie die Datei zu klingen hat und **berechnet die Änderungen in Echtzeit beim Abhören**. Möchtest Du die **Änderungen festschreiben**, so musst Du die Datei **exportieren (bouncen)** – in eine neue Datei. Dein **Original bleibt jederzeit unangetastet** und Du kannst schnell zum Ursprung zurückkehren.

Vorteile non-destruktiv

- Das Original ist jederzeit sicher
- Du kannst immer wieder von vorne beginnen ...
- und ganze neue Effekte ausprobieren
- Fehler lassen sich leicht ausmerzen

Vorteile destruktiv

Selbst wenn die Audiorechner heute eine Menge Leistung bereithalten – sie ist endlich. Und spätestens hier kommt die destruktive Audiotbearbeitung ins Spiel. Schreibst Du die Effekte in die Datei fest (destruktiv), wird keine zusätzliche Rechenleistung für die Berechnung der Effekte mehr benötigt. Diese Audiodatei wird von allen Geräten und Musikprogrammen in exakt demselben Zustand abgespielt.

Effekt: Fade-In & Fade-Out (Einblenden & Ausblenden)

Fade-Ins und Fade-Outs werden dazu genutzt, das Audiosignal **sanft ein- bzw. auszublenden**. Das kennst Du beispielsweise aus dem Radio, wo ein Song reinkommt oder wieder rausgefahren wird, während die Moderation startet.

Wenn bei einer Aufnahme die Welle nicht mit einem Nullpunkt beginnt, führt dies zu einem mehr oder minder leisen Knackser. Auch an den Übergängen zweier Audioclips kann es zu einem solchen Knacksen kommen. Eine sehr kurze Einblendung, ein Fade-In zu Beginn und eine Ausblendung, ein Fade-Out am Ende sorgen hier für Nullpunkte und damit knackserfreie Übergänge.

Viele Musik- und Audiotbearbeitungsprogramme machen das inzwischen automatisch am Beginn und am Ende einer Audiodatei. Die Handbücher weisen dieses Feature als »automatic fade«, »Auto-Fade«, »automatische Ein-/Ausblendung« oder dergleichen aus.

Tempo ändern

Das Tempo, auch Zeitmaß, gibt in der Musik an, wie schnell ein Stück zu spielen ist, bestimmt also die absolute Dauer der Notenwerte.

Tonhöhe ändern

Die Tonhöhe wird in der Praxis mit der **Frequenz (Schwingungsanzahl pro Zeitspanne) eines hörbaren Tones** gleichgesetzt, genauer: mit der **Grundfrequenz**, also der Frequenz des tiefsten Teiltons in dem betreffenden Klang. Die Tonhöhe ist neben der Tondauer, der Lautheit und der Klangfarbe eine wichtige Eigenschaft musikalischer Töne und der Vokale gesprochener Sprache.

Lautstärke ändern

Die Lautstärke ist die physikalisch messbare Amplitude oder Stärke des Schalls (z. B. als Schalldruck bzw. als Schalldruckpegel).

Audioformate

Die Vielzahl der digitalen Audio ist für den Nutzer manchmal verwirrend. Hier findest du eine Übersicht der gängigsten Audioformate und deren Dateiendungen inklusive einer Kurzbeschreibung.

.WAV - wave form audio

Das WAV Dateiformat dient der digitalen Speicherung von Audiodaten. In den allermeisten Fällen enthält eine WAV Datei dabei unkomprimierte Rohdaten. Der Verzicht auf eine Komprimierung bedeutet folglich, dass die Audiodatei in höchster Qualität gespeichert ist. Allerdings erfordert dies einen enormen Speicherbedarf. Zwei Minuten Musik können schnell um die 20 Megabyte (MB) Platz belegen.

.MP3 - MPEG-1 Audio Layer 3

Von einer Expertengruppe entwickeltes Audioformat mit standardisierten Komprimierungsverfahren. Die Vorteile von MP3 Audiodateien sind die vergleichsweise kleinen Dateien bei dennoch sehr guter Qualität. In der Praxis bedeutet dies, dass Sie den Inhalt einer kompletten Musik CD ohne wahrnehmbare Qualitätsverluste auf die Größe von 10 bis 20 Megabyte (MB) reduzieren können.

.WMA - Windows Media Audio

Ein von Microsoft © entwickeltes Audioformat für Windows Betriebssysteme. Hohe Kompression mit guter Qualität. Das bedeutet relativ kleine Dateien und dennoch Musik- oder Sounddateien mit guter Qualität möglich.

.AAC - Advanced Audio Coding

Ein ebenfalls von der MPEG (Moving Picture Experts Group) entwickeltes Audiodatenkompressionsverfahren. AAC gilt als qualitativ verbesserter Nachfolger von MP3. Die Vorteile von AAC Audiodateien sind die vergleichsweise kleinen Dateien bei dennoch exzellenter Qualität. In der Praxis bedeutet dies, dass Sie den Inhalt einer kompletten Musik CD ohne wahrnehmbare Qualitätsverluste auf die Größe von 10 bis 20 Megabyte (MB) reduzieren können.

OGG - Ogg Vorbis genannt

Ein freier (patentfreier) Codec zur verlustbehafteten Audiodatenkompression, der von der Xiph.Org Foundation als Alternative zum weit verbreiteten MP3 Format entwickelt wurde.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:03

Last update: **2019/05/14 15:41**



Shortcuts in Audacity

Leertaste	Abspielen - Play/Stop
F1	Auswahlwerkzeug
F2	Hüllkurvenwerkzeug - Lautstärke verstellen
F3	Zeichenwerkzeug - einzelne Samples verändern
F4	Zoomwerkzeug / ev. mit Shift
Strg+Mausrad	Zoomen
F5	Bewegen von Auswahlbereichen
Strg+d	Duplizieren des Auswahlbereichs auf neue Spur
Strg+x	Ausschneiden
Strg+Alt+x	Ausschneiden, Spur rückt nicht nach
Strg+v	Einfügen
Strg+t	Nur Auswahlbereich übrig lassen
Entf	Entfernen
Strg+Alt+k	Löschen, Spur rückt nicht nach
Strg+l	in Stille umwandeln
Strg+i	Trennen

Rauschentfernung

- Effekt - Rauschentfernung
 - zuerst Rauschprofil ermitteln (Teil, wo Rauschen auftritt)
 - Rauschentfernung

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:03:shortcuts_audacity



Last update: **2019/05/10 13:13**

Übungen

- [Übung Sprechpuppe](#)
- [Übung Audioschnitt](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:03:uebungen



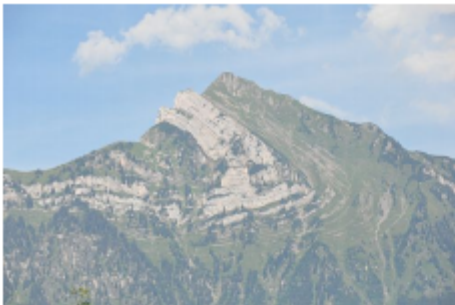
Last update: **2019/05/10 13:14**

4) Digitale Bildbearbeitung

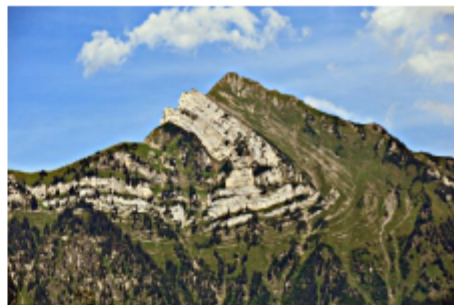
4.1) Theorie

Die Bildbearbeitung dient zur Veränderung von digitalen Bildern. Oft wird die Bildbearbeitung dazu angewandt, um Fehler (z.B. Rote Augen, zu viel/wenig Helligkeit, ...) zu beseitigen, die beim Fotografieren entstehen. Bildbearbeitung wird auch kreativ angewandt, um Fotos zu verändern oder neu zusammenzusetzen.

Die folgenden Bilder zeigen einige Möglichkeiten der Bildbearbeitung:



wenig Kontrast, flau



Kontrast erhöht und nachgeschärft



Altes Foto: ausgebleicht mit Schäden



Schäden retuschiert



Original



Effektfilter angewendet



Himmel durch eine Grafik ersetzt



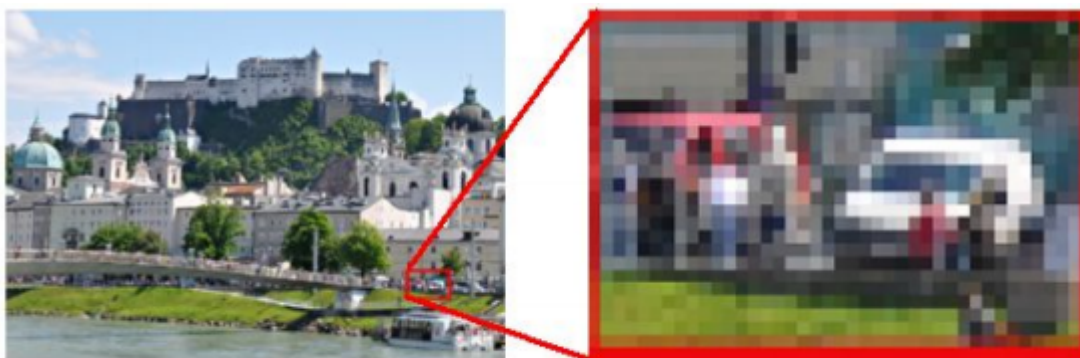
Lichteffekt, Beschriftung eingefügt

Rastergrafik (Pixelgrafik oder Bitmaps) vs. Vektorgrafik

Man unterscheidet grundsätzlich zwischen diesen beiden Arten von Grafiken.

Rastergrafiken / Pixelgrafiken

bestehen aus Bildpunkten (Pixel), denen Farbinformation zugeordnet sind. Die Größe der Grafik hängt von der Anzahl und Größe dieser Farbpunkte ab. Bei der Bearbeitung von Pixelgrafiken werden die einzelnen Bildpunkte bearbeitet. Rastergrafiken werden für Digitalfotos und eingescannte Bilder verwendet. Rastergrafiken können nicht beliebig vergrößert werden: je stärker die Vergrößerung, desto mehr erkennt man die Pixelstrukturen (es wird pixelig).



Pixelstrukturen werden im vergrößerten Bildausschnitt sichtbar

Dateiformate für Rastergrafiken

JPG	Grafikformat zur Speicherung von Bildern mit verlustbehafteter Kompression. Als Dateinamenerweiterung wird meistens jpg , seltener jpeg oder jpe verwendet. Die meisten Fotos im Internet werden im Format JPG gespeichert.
PNG	Grafikformat mit verlustfreier (lossless) Kompression. PNG unterstützt neben unterschiedlichen Farbtiefen auch Transparenz per Alphakanal. PNG ist das meistverwendete verlustfreie Grafikformat im Internet.
GIF	Grafikformat mit verlustfreier Kompression für Bilder mit bis zu 256 Farben. GIF-Bilder können auch Transparenz darstellen. GIF-Bilder können aus mehreren Einzelbildern bestehen, die als Animation abgespielt werden.
TIF	Professionelles Grafikformat für Austausch von hochwertigen Bilddateien. TIFF speichert mit verlustfreier Kompression und erlaubt hohe Farbtiefen.
BMP	Unkompromiertes Grafikformat mit hoher Farbtiefe. Aufgrund des hohen Speicherbedarfs nicht für das Internet geeignet.

Vektorgrafiken

Eine Vektorgrafik ist eine Computergrafik, die aus Objekten wie Linien, Kreisen, Polygonen oder Kuren zusammengesetzt ist. Um beispielsweise das Bild eines Kreises zu speichern, benötigt eine Vektorgrafik mindestens zwei Werte:

- die Lage des Kreismittelpunkts und
- den Kreisdurchmesser

Neben der Form und Position der Objekte werden auch die Farbe, Strichstärke, diverse Füllmuster und weitere, das Aussehen bestimmende Daten angegeben.



Vektorgrafiken können ohne Qualitätsverlust vergrößert werden. Viele Vektorgrafiken findet man auf <https://www.freepik.com/free-vectors/graphics>

Grafikformate für Vektorgrafiken:

SVG	Scalable Vector Graphics (SVG, engl. „skalierbare Vektorgrafik“) ist die vom World Wide Web Consortium (W3C) empfohlene Spezifikation zur Beschreibung zweidimensionaler Vektorgrafiken. Aktuelle Webbrowser können SVG-Grafiken darstellen. SVG-Dateien bestehen aus Text und können mit einem Texteditor bearbeitet werden.
EPS	Das Vektorformat Encapsulated PostScript (EPS) wurde von Adobe entwickelt. Es wird von zahlreichen Programmen unterstützt und im professionellen Bereich häufig als Austauschformat für Vektorgrafik (und Bitmaps) verwendet.

Komprimierung von Bildern

Nicht komprimierte Bilder benötigen zur zur Speicherung sehr viel Speicherplatz. Um den Speicherbedarf zu verringern, werden Bilddateien komprimiert. Man unterscheidet zwischen der verlustfreien und der verlustbehafteten Kompression:

Verlustfreie (lossless) Kompression

Bei der verlustfreien Kompression geht keine Information verloren. Die Daten werden nur anders als vorher organisiert, indem bestimmte Redundanzen (Wiederholungen) erkannt und zusammengefasst werden. Aus den komprimierten Daten kann das Originalbild zu 100% wieder rekonstruiert werden. Dateiformate für verlustfreie Kompression sind **GIF, PNG, TIF und JPEG 2000**.

Verlustbehaftete Kompression

Bei der verlustbehafteten Kompression wird versucht, den Informationsverlust unmerklich oder wenigstens erträglich zu halten. Bei der Komprimierung werden Bildinformationen entfernt, die für das Auge kaum wahrnehmbar sind. Je stärker das Bild komprimiert wird, desto deutlicher werden Bildfehler erkennbar. Das am häufigsten eingesetzte Dateiformat für verlustbehaftete Kompression ist **JPEG**

Kompressionsverfahren im Vergleich

Die verbreiteten Formate weisen Vor- und Nachteile auf und eignen sich daher für den jeweiligen Zweck mehr oder weniger gut. Computergrafiken, die harte Kanten enthalten – etwa einfache Grafiken oder Screenshots – können durch verlustfreie Verfahren meist besser komprimiert werden. So zeigt das folgende Beispiel, dass bei gleicher Dateigröße JPG deutliche Fehler (Artefakte) aufweist.



Das JPG-Format eignet sich besonders gut zur Speicherung von Fotos, wenn es auf geringen Speicherbedarf ankommt. Verlustfreie Formate wie PNG eignen sich in der Regel nicht für Fotografien, da sie erheblich größere Dateien produzieren.

Farbmodelle

Licht besteht aus elektromagnetischen Wellen. Die Wellenlänge bzw. die Frequenz der Lichtwelle bestimmt die Farbe des Lichts. Der Mensch hat zur Farberkennung drei verschiedene Farbrezeptoren, die Zapfen, die unterschiedliche Empfindlichkeit für Wellenlängen aufweisen. Dadurch kann der Mensch verschiedene Farben unterscheiden. Man unterscheidet verschiedene Farbmodelle:

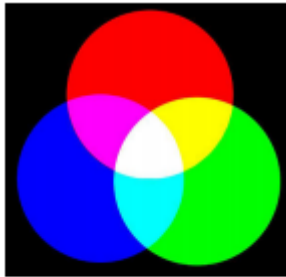
RGB-Farbmodell

Im RGB-Farbmodell (RGB: **R**ot, **G**rün, **B**lau) werden Farben durch Mischen der drei Grundfarben Rot, Grün und Blau dargestellt. Der RGB-Farbraum ist ein additives Farbmodell: Farben werden durch ausgesendetes Licht dargestellt. Je nach Anteil von rotem, grünem und blauem Licht werden verschiedene Farben wiedergegeben.

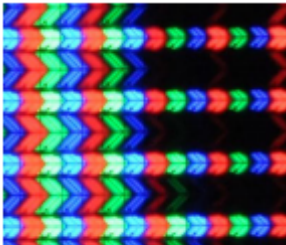
Man kann sich die Farbkreise im Bild rechts als drei Scheinwerfer vorstellen, die eine rote, eine grüne und eine blaue Kreisfläche an die Wand projizieren. Dort wo alle drei Farben übereinander liegen, ergibt sich Weiß. Die Addition von Rot und Grün ergibt die Farbe Gelb etc.

Pro Farbe stehen 256 Helligkeitsstufen zur Verfügung. Ist jede Farbe auf maximale Helligkeit (255) eingestellt, so ergibt sich die Farbe Weiß, bei einer Einstellung von 0 erhält man Schwarz.

Daher hat das RGB-Modell eine Farbtiefe von 24 Bit (8 Bit pro Kanal) und kann 16,7 Mio. Farben darstellen. Die restlichen Farben werden durch Farbaddition der Grundfarben dargestellt; d.h. alle Farben zusammen ergeben nicht Schwarz, sondern heben sich auf und ergeben Weiß. Bildschirme gehen immer von einem schwarzen Hintergrund aus, auf dem durch Aktivieren von Elektronenstrahlen rote, blaue und grüne Punkte zum Leuchten gebracht werden. Für jeden Bildpunkt stehen drei verschiedenfarbige Pixel zur Verfügung, deren Leuchtstärke je nach erwünschter Farbe geändert wird.



RGB-Farbmodell



LCD-Fernseher: stark vergrößerter Ausschnitt

CMYK - Farbmodell

CMYK steht für die Farben **C**yan, **M**agenta, **Y**ellow und Schwarz (**K**ey). Diese Farben stehen für Druckfarben und ihr Anteil wird in Prozent von 0 bis 100 % angegeben.

Das CMYK-Farbmodell ist ein **subtraktives** Farbmodell: die im Bild rechts gezeigten Farbscheiben kann man sich als bunte Filterscheiben vor einem weißen beleuchteten Hintergrund vorstellen.

Die Mischung der Farben Cyan, Magenta, Yellow ergibt nur theoretisch Schwarz, in der Praxis wird im Druckprozess Schwarz (**K**ey) hinzugefügt. Dabei wird zusätzlich noch ein besserer Kontrast erzielt.

Alle Farbdrucker haben mindestens CMY-Farben und Schwarz. Jeder Druckvorgang eines Bildes im RGB-Farbmodell erfordert vor dem Ausdruck eine Umrechnung in das CMYK-Farbmodell.



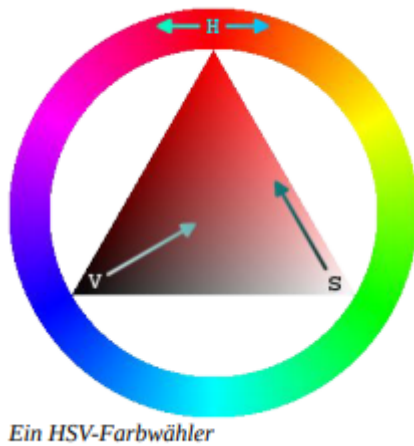
CMYK-Farben

HSV - Farbmodell

Im HSV-Farbraum bestimmt man eine Farbe mit Hilfe des Farbwerts (englisch **h**ue), der Farbsättigung (**s**aturation) und des Helligkeitswerts (**v**alue).

- **Farbwert** als Farbwinkel H auf dem Farbkreis (etwa 0° für Rot, 120° für Grün, 240° für Blau)
- **Sättigung** S in Prozent (0 % = Neutralgrau, 50 % = wenig gesättigte Farbe, 100 % = gesättigte, reine Farbe) oder in einem Intervall von Null bis Eins
- **Hellwert** V als Prozentwert (0 % = keine Helligkeit, 100 % = volle Helligkeit), oder in einem

Intervall von Null bis Eins, auch Dunkelstufe genannt.



Graufen - Farbmodell

Ein Graustufenbild hat keine Farben. Es besteht aus verschiedenen Grauwerten in Abstufungen von 0 (Schwarz) bis 255 (Weiß).

Farbtiefe

Bilder bestehen aus Bildpunkten, denen Farben zugeordnet sind. Die Farbtiefe gibt an, wie viele Farbabstufungen möglich sind und wird in Bit angegeben. Je größer die Anzahl der Bits ist, desto mehr Farbabstufungen sind möglich.

Bild	Farbtiefe	Anzahl Farbabstufungen
Schwarz-Weiß-Bild ohne Graustufen	1 Bit	2 Farben: Schwarz und Weiß
Graustufenbild	8 Bit	$2^8 = 256$ Farben
GIF-Bild (indizierte Farben)	8 Bit	$2^8 = 256$ Farben
Farbbilder (JPG)	24 Bit = 3 Farben je 8 Bit	$2^{24} = 16,7$ Mill. Farben
Farbbilder mit Transparenz (TIF)	32 Bit = 3 Farben + Transparenz je 8 Bit	$2^{32} = 4,3$ Mrd. Abstufungen
CMYK	32 Bit = 3 Farben + K-Wert je 8 Bit	$2^{32} = 4,3$ Mrd. Farben

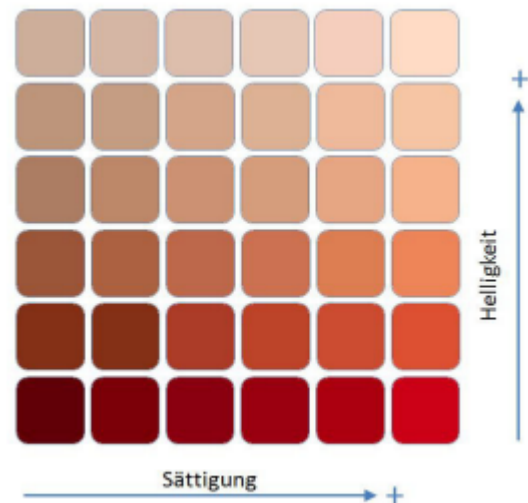
Farbpaletten - indizierte Farben

Im GIF-Format kann man pro Grafik aus den 16,7 Mio. möglichen Farben des RGB-Farbmodells maximal 256 Farben wählen und abspeichern. Bei einer solchen Farbauswahl spricht man von einer Farbpalette. Jede der 256 Farben der Farbpalette bzw. Farbtabelle kann beliebige RGB-Werte haben. Zu jedem Bildpixel wird also nur die Nummer der dazugehörigen Farbe aus der Farbpalette gespeichert. Dadurch wird deutlich Speicherplatz gespart.



Farbton

Der Farbton bezeichnet in der Farbenlehre die Eigenschaft, nach der man Farbempfindungen nach beispielsweise **rot**, **gelb** oder **grün** unterscheidet. Eine Farbe desselben Farbtons kann entweder in der **Farbsättigung** variieren, wie graublau oder in der **Helligkeit**, beispielsweise rosa.



Farbsättigung

Die Farbsättigung gibt die Stärke bzw. Intensität der Farbe an: Je bunter und kräftiger uns die Farben eines Bildes erscheinen, desto höher ist die Farbsättigung. Flaue Farben haben eine geringe Farbsättigung. Bilder, die entsättigt wurden, haben nur noch Grautöne.



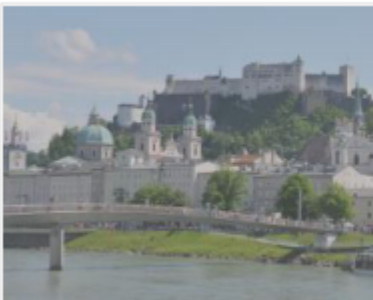
Farbbalance

Ein Bild mit guter Farbbalance zeigt keinen Farbstich. Die Beurteilung der Farbbalance eines Bildes ist subjektiv und hängt von Geschmack und Urteilsvermögen ab. Im Bild rechts ist deutlich ein Farbstich zu sehen – hier wurde die Farbbalance korrigiert.



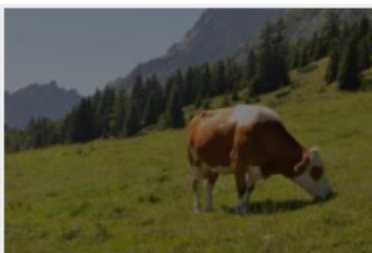
Kontrast und Dynamik

Der **Kontrast** bezeichnet den Unterschied zwischen hellen und dunklen **Bereichen** eines Bildes. Der **Kontrastumfang** oder die **Dynamik** beschreiben den **Intensitätsunterschied** zwischen dem **hellsten** und **dunkelsten Punkt** eines Bildes.



Helligkeit (Luminanz)

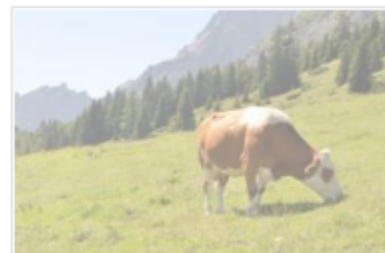
Die **Luminanz** ist eine fotometrische Größe aus der Videotechnik, die als Maß für die Helligkeit von Bildpunkten verwendet wird. Jede Farbe im RGB-Farbmodell kann 256 Helligkeitsstufen annehmen. Verändert man bei einem Bild die Helligkeit, so wird bei allen Pixeln die Helligkeitsstufe erhöht bzw. verringert. Bei einem Foto führt die Änderung der Helligkeit meist zu keinem befriedigenden Ergebnis: schwarze Bereiche werden heller bzw. weiße Stellen werden hellgrau. Damit schwarze Bereiche schwarz und weiße weiß bleiben, muss man eine **Gammakorrektur** durchführen!



verringerte Helligkeit



normale Helligkeit



erhöhte Helligkeit

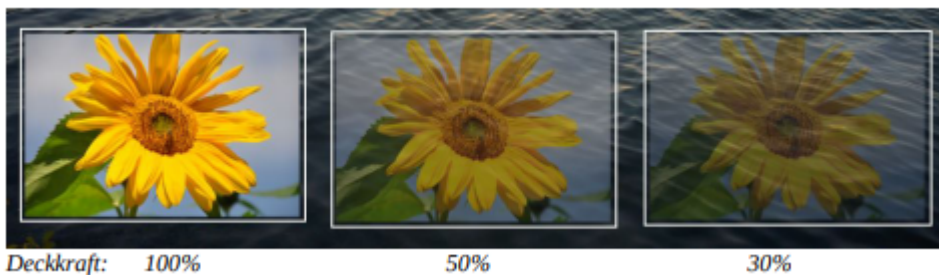
Gamma

Bei einer Gammakorrektur bleibt Schwarz auch schwarz und Weiß bleibt weiß – nur die mittleren Farbtöne werden verändert.



Um die „Helligkeit“ eines Fotos zu ändern ist so gut wie immer eine Gammakorrektur der richtige Weg!

Transparenz



Die Transparenz ist die Durchsichtigkeit bzw. Lichtdurchlässigkeit einer Fläche. In der Bildbearbeitung werden oft Bildebenen verwendet. Durch transparente Flächen scheinen darunter liegende Bildflächen durch. Die **Deckkraft** bestimmt, wie durchsichtig die Fläche ist. Sie geht von 100 % (keine Transparenz) bis 0 % (komplette Transparenz). Dateiformate zur Speicherung von Transparenz sind beispielsweise **PSD** (Photoshop), **XCF** (Gimp), **PNG** und **TIF** (stufenlose Transparenz) und **GIF** (nur eine transparente „Farbe“). **JPG** kennt keine Transparenz

4.2) Praxis mit Adobe Photoshop

- [Skriptum Photoshop](#)
- [Grundeinstellungen](#)
- [Ebenen](#)
- [Auswahlwerkzeuge](#)
- [Bildqualität optimieren](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:04

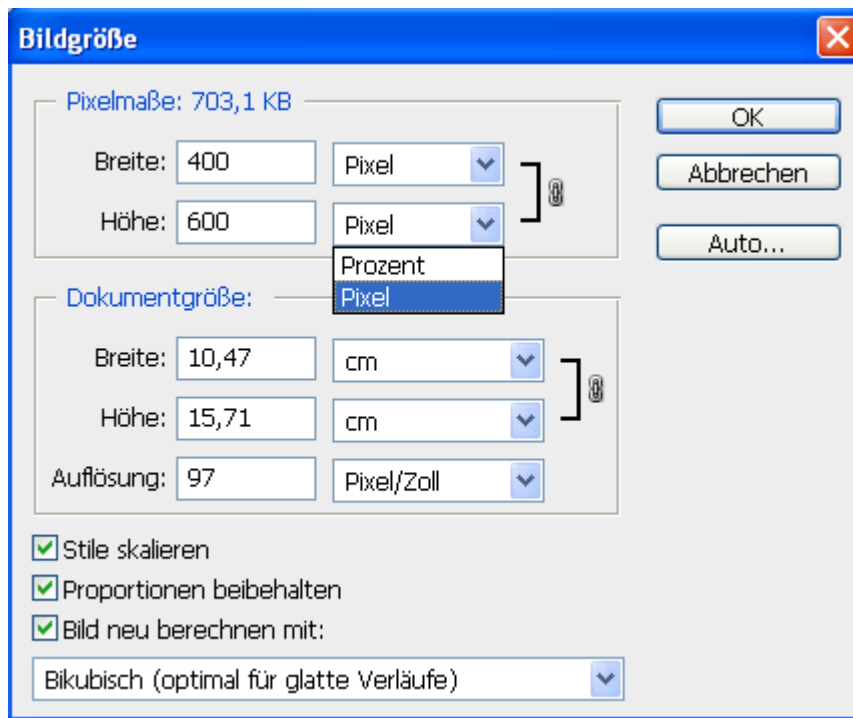


Last update: **2019/05/14 15:42**

Einstellungen der Bildgröße und Bildabmessungen

Auflösung, Bildgröße und Farbtiefe ändern

Um die Bildgröße, Auflösung oder Farbtiefe zu ändern, wählt man den Menüpunkt Bild - Bildgröße.



Im Dialogfeld **Pixelmaße** wird die Größe des Bildes in Pixel angezeigt. Dort kann man auch die gewünschte Größe in Pixel angeben. Darüber hinaus kann man das Bild mit Hilfe des Listenfeldes **Prozent** auch prozentual vergrößern oder verkleinern.

Ist das Kontrollfeld **Proportionen erhalten** aktiv, so wird bei der Eingabe des einen Wertes, der zweite Wert automatisch ermittelt.

Auflösung

Mit der Auflösung wird festgelegt, wie viel Pixel pro Zentimeter auf einem Ausgabemedium wiedergegeben werden. Da viele Bildbearbeitungen aus den USA stammen, wird für die Definition der Auflösung überweise das amerikanische Längenmaß Inch (Zoll=2,54 cm) verwendet.

Wie bereits erwähnt, ist die Auflösung abhängig vom Ausgabemedium. Es gilt das Prinzip: *So viel wie nötig, so wenig wie möglich*, da sich die Wahl der Auflösung deutlich auf die Dateigröße auswirkt. Im Dialogbereich **Dokumentgröße** bzw. **Dateigröße** kann die Bildgröße in absoluten Werten (cm, mm, pt,...) festgelegt werden.

Weiters kann dort auch die Auflösung festgelegt werden.

Eine zu hohe Auflösung führt zu einer großen Dateigröße, eine zu geringe Auflösung sorgt für einen

unschönen Mosaik-Effekt.

Bilder beschneiden

Das Beschneiden von Bildern kann beispielsweise auf zwei Arten erfolgen:

- Auswahlwerkzeug + Menüpunkt: Bild - Freistellen
- Freistellungswerkzeug

Bild beschneiden mit dem Menüpunkt BILD - FREISTELLEN

- Wähle in der Toolbox ein Auswahlwerkzeug (z.B. Rechtecke)
- Zeichne einen Auswahlrahmen um den Bereich, der erhalten werden soll.
- Wähle BILD - FREISTELLEN

Bild beschneiden mit dem Freistellungswerkzeug

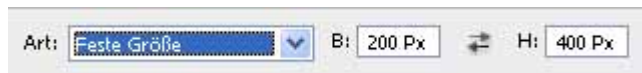
Mit dem Freistellungswerkzeug können Bilder beschnitten werden, indem man einen Rahmen aufzieht. Um einen quadratischen Rahmen zu erhalten muss man die Umschalt-Taste gedrückt halten.

Ist ein Auswahlbereich festgelegt, kann dieser nachträglich noch vergrößert, verschoben oder auch gedreht werden. Um das Bild endgültig freizustellen klickt man doppelt innerhalb des ausgewählten Bereichs oder aktiviert in der Optionsleiste das Symbol ✓.

Bild auf eine bestimmte Größe zuschneiden

Für bestimmten Anwendungen, beispielsweise fürs Web, ist es manchmal notwendig, eine Bild auf eine bestimmte Größe zuzuschneiden.

Das empfiehlt sich die Anwendung von Varianten 1 (Menüpunkt Bild-Freistellen).



Stellt man in der Optionsleiste die Art der Auswahl auf Feste Größe (s. Abb. oben), so können die gewünschten Werte für Breite und Höhe in cm oder Pixel (px) eingegeben werden. Der Auswahlrahmen nimmt dann automatisch die vorgegebene Größe an.

Aufgabe 1

Öffne die Datei „birgit.png“. Das Bild soll auf die Größe von 350 px mal 450 px zugeschnitten werden! Das Gesicht der Person soll erhalten werden.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:04:04_01



Last update: **2019/05/10 14:07**

Ebenen

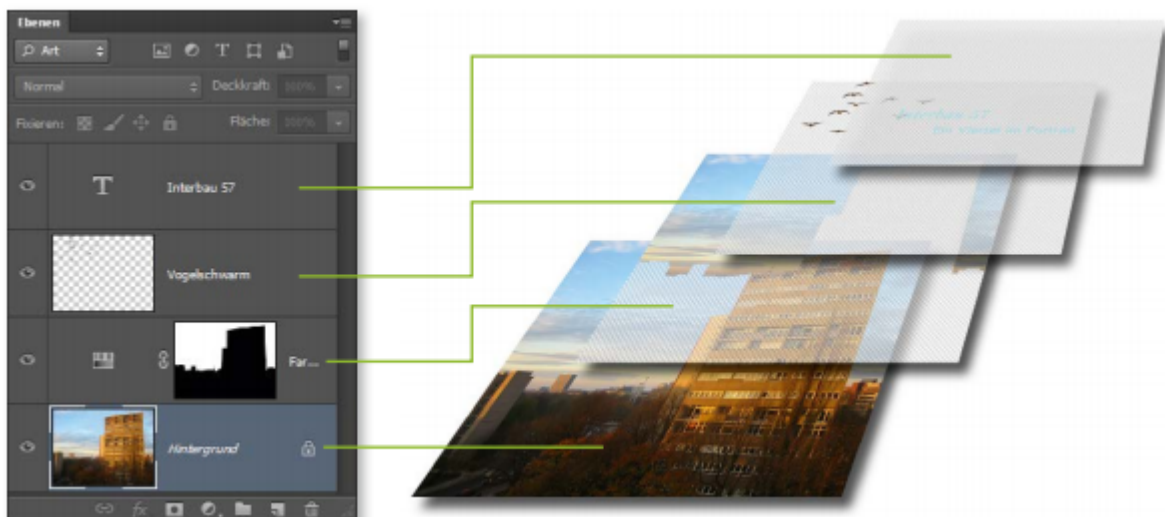
Jedes Bild, das Sie in Photoshop öffnen oder neu anlegen, besteht aus mindestens einer Ebene. Weitere Ebenen können Sie in fast unbegrenzter Zahl hinzufügen. Die Grenze des Machbaren wird somit weniger vom Programm als von der Kapazität Ihres Rechners bestimmt, denn die Größe einer Datei und die Rechenzeit für einzelne Arbeitsschritte steigen rapide an, je mehr Ebenen vorhanden sind.

Vorteile

- Ebenen ermöglichen das separate Bearbeiten, Verschieben, Kopieren, Verändern und Korrigieren einzelner Bildteile
- ein einfaches Anbringen von Änderungen auch bei komplexen Kompositionen
- das Herstellen von Bildvarianten
- kreatives Experimentieren.

Funktionsweise

Stellen Sie sich Ebenen wie übereinandergeschichtete Folien vor. Jede der Folien ist ganz oder teilweise mit Pixeln gefüllt, und auch die Deckkraft von Pixeln auf einer Ebene lässt sich stufenlos ändern. Ebenen mit so verringerter Deckkraft oder nur teilweise mit Bildpixeln gefüllte Ebenen (im Beispielbild die Ebene »Vogelschwarm«) lassen die Inhalte darunterliegender Ebenen erkennen (im Beispiel die Hintergrundebene). Die Reihenfolge der Ebenen im Ebenen-Bedienfeld entspricht der Schichtung der Ebenen im Bild und ist für das Aussehen des Gesamtbildes maßgeblich. Zudem können Sie festlegen, ob und wie die Pixel übereinanderliegender Ebenen miteinander verrechnet werden.



Ebenentransparenz und Ebenendeckkraft

Einem reinen Bild sehen Sie die Ebenen, aus denen es zusammengesetzt ist, nicht an – und auch nicht immer die Deckkraft der Bildpixel. Erst ein Blick in das Ebenen-Bedienfeld offenbart die Ebenenstruktur.

Ebenentransparenz

In der digitalen Fotografie ist Transparenz die Funktionalität, die transparente Bereiche in einem Bild oder einer Bildebene unterstützt. Bestimmte Bildformate bieten keine Unterstützung für die Transparenz. Die Deckkraft ist die Fähigkeit eines Materials bzw. Objekts, Licht nicht durchzulassen. Sie können die Deckkraft von Ebenen, Filtern und Effekten ändern, um mehr oder weniger des jeweils darunter liegenden Bildes durchscheinen zu lassen.

Ebenendeckkraft

Es ist auch möglich, die Deckkraft von Bildpixeln einer Ebene herabzusetzen. Liegt unter der deckkraftreduzierten Ebene eine weitere Bildebene, wird diese sichtbar. Liegt unterhalb der Ebene, deren Deckkraft gesenkt wurde, keine weitere Ebene, ist wieder das grau-weiße »Schachbrett« (=Transparenz) zu sehen.



◀ **Abbildung**
Hier liegen die Buddha-Figur und ein hellblauer Hintergrund auf zwei getrennten Ebenen. Die Ebene »Blauer Hintergrund« ist in der Deckkraft 1 auf 70 % reduziert. Das Schachbrettmuster ist daher ein wenig zu sehen.

Ebenenarten

In Photoshop arbeiten Sie mit verschiedenen Ebenenarten, die sich hinsichtlich möglicher Inhalte, Bearbeitungsmöglichkeiten und Einsatzzwecke voneinander unterscheiden.

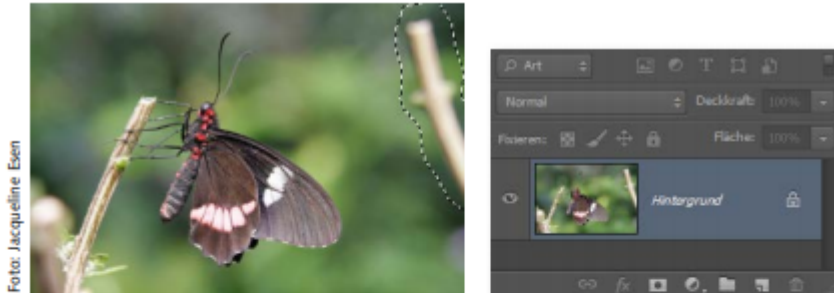
Bildebenen

»Normale« Bildebenen sind der mit Abstand am häufigsten genutzte Ebenentyp. Bildebenen enthalten Pixelinformationen oder Transparenz und lassen sich mit allen Funktionen und Werkzeugen bearbeiten.

Hintergrundebenen

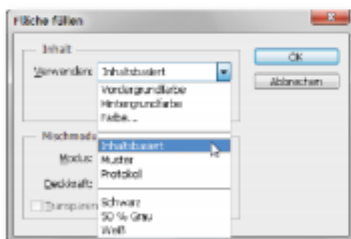
Die Hintergrundebene ist immer die unterste Ebene einer Datei. Sie erkennen sie auch am kursiv geschriebenen Ebenentitel Hintergrund. Pro Bild kann es nur eine Hintergrundebene geben. Hintergrundebenen unterscheiden sich in einigen Details von normalen Bildebenen: Sie können nicht transparent sein, und nicht alle Arbeitstechniken sind auf sie anwendbar. So können Sie Deckkraft

und Mischmodus von Hintergrundebenen nicht verändern, können keine Ebenenstile zuweisen, und auch beim Löschen von Pixeln gibt es eine Besonderheit (siehe unten). Außerdem lassen sich Hintergrundebenen nicht transformieren. Gedacht sind sie wohl als eine Art »Mal-Leinwand«. Es ist jedoch auch möglich, Bilder ganz ohne Hintergrundebene, ausschließlich mit anderen Ebenenarten, zu erstellen. Allerdings haben einige andere Anwendungen Schwierigkeiten, Dateien ohne reguläre Hintergrundebene zu verarbeiten



▲ Abbildung

Das Dokument besteht aus einer Hintergrundebene, der ausgewählte Bereich soll gelöscht werden (etwa durch Betätigen der **[Entf]**-Taste).



▲ Abbildung

Unter **VERWENDEN** steht auch die Option **INHALTSBASIERT** zur Verfügung. Ist sie aktiv...



▲ Abbildung

...berechnet Photoshop neue Pixel für den weggeschnittenen Bereich, die zur Umgebung passen. Je nach Motiv funktioniert das mehr oder weniger gut.

Textebenen

Textebenen erkennen Sie an dem großen »T« in der Ebenenminiatur. In ersten Abbildung ist die oberste Ebene eine Textebene (»Interbau 57«).

Wenn Sie Text in ein Bild einfügen, besteht diese Schrift aus Pixeln – bei vergrößerter Bildansicht werden die typischen zackigen Pixelkanten sichtbar. Tatsächlich besteht Text in Photoshop jedoch aus mathematisch definierten Formen (Vektoren!), die die einzelnen Zeichen einer Schrift beschreiben. Dadurch sind Schriften verlustfrei skalierbar und ergeben trotz der »pixeligen« Bildschirmdarstellung ein scharfes Bild im Druck – vorausgesetzt, Dateiformat und Drucker stimmen

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:04:04_02



Last update: **2019/05/10 14:21**

Auswahlwerkzeuge

Rechteck und Ellipse

Der Einsatz des Auswahlwerkzeugs „Rechteck und Ellipse“ wurde bereits im vorangehenden Kapitel [Einstellung der Bildgröße und Bildabmessung](#) erläutert.

Eine andere Möglichkeit um die Optik ausgeschnittener Bildteile zu verbessern, ist die Verwendung „weicher Kanten“. Im Optionsfeld oder mit Hilfe des Kontextmenüs kann im Feld **weiche Kante** ein Wert eingegeben werden. Dabei werden die Pixel des Kantenbereichs abgestuft und erscheinen weicher bzw. etwas verschwommen. Empfehlenswert für die Weiterverarbeitung ist es, diese Bildabschnitte anschließend auf einen transparenten Hintergrund einzufügen.

Aufgabe

- Hole dir die Datei Blume von Moodle!
- Du möchtest die „rote Blume“ auf das Titelblatt einer Präsentation einfügen und benötigst deshalb nur einen Bildausschnitt.
- Markiere das Bild mit dem Auswahlwerkzeug „Ellipse“ und gib bei „weiche Kante“ den Wert 20px ein.
- Kopiere den Bildausschnitt in eine neue Datei mit transparenten Hintergrund und speichere das Bild unter „Blume_weicheKante.png“.
- Jetzt wiederhole den Vorgang, wobei du den Wert weiche Kante zurück auf 0 px setzt („Blume_harteKante.png“)!
- Öffne eine leere Powerpoint-Präsentation und formatiere diese mit einer passenden Hintergrundfarbe!
- Anschließend füge beide Bilder ein und vergleiche das Ergebnis!

Lasso-Werkzeuge

Das Lasso-Werkzeug

Ein Werkzeug zur Auswahl von unregelmäßigen und schlecht konturierten Bildbereichen ist das Lasso-Werkzeug (bzw. Freihand-Lasso). Dabei werden die auszuwählenden Bildbereiche freihändig mit dem Lasso umkreist.

Aufgabe

Öffne die Datei „Apfel“ und stelle den Apfel mit Hilfe des „Lasso-Werkzeuges“ frei!

Das Polygon-Lasso-Werkzeug

Der große Vorteil des Polygon-Lassos gegenüber dem Standard-Lasso ist, dass während der Erstellung der Auswahl die Maustaste nicht permanent gedrückt werden muss. Stattdessen wird die Auswahl anhand eines Linienzugs definiert, der Punkt für Punkt erstellt wird.

Tipps

- Während der Erstellung eines Linienzugs kann der zuletzt gesetzte Punkt mit der ENTF - Taste gelöscht werden.
- Die gesamte Arbeit mit dem Polygon-Lasso kann mit Hilfe der ESC-Taste abgebrochen werden.
- Besonders empfehlenswert ist es, vor der Arbeit mit dem Lasso in das Bild zu zoomen, um eine möglichst genaue Auswahl treffen zu können.

Besonders bei der Arbeit mit dem Lasso, erscheint der erzeugte Linienzug oft sehr eckig. Um dies zu verbessern, sollte man bei der Arbeit mit dem Polygon-Lasso möglichst viele Punkte bei z.B. Rundungen setzen.

Aufgabe

Öffne die Datei „Gabel.psd“. Ziehe mit Hilfe des Polygon-Lassos einen Linienzug rund um die Gabel und kopiere die Gabel anschließend in eine neue Datei.

Magnetisches Lasso-Werkzeug

Das dritte Lasso-Werkzeug ist das magnetische Lasso. Dieses Werkzeug reagiert magnetisch auf im Bild vorhandene Kanten und sollte zur Erstellung von Auswahlen rund um gut konturierte Kanten innerhalb eines Bildes verwendet werden.

Der erste Klick mit dem magnetischen Lasso sollte genau die gewünschten Kanten treffen, danach genügt es, den Mauszeiger in etwa entlang der Kante zu bewegen. Es werden währenddessen automatisch Punkte auf der von Photoshop erkannten Kante erzeugt. Die Auswahl wird durch einen Doppelklick oder durch einen Klick auf den zuerst erzeugten Punkt beendet.

Die Werkzeugoptionen des magnetischen Lassos bieten neben den auch bei anderen Lasso-Werkzeugen vorhandenen Optionen noch drei zusätzliche Punkte:

- Die **Breite** bestimmt den Bereich rund um die aktuelle Mausposition, der für die Kantenerkennung verwendet wird.
- Mit dem **Kantenkontrast** wird der für eine Kante notwendige Tonwertunterschied festgelegt.
- Die **Frequenz** legt fest, wie häufig automatische Punkte gesetzt werden.

Aufgabe

Öffne die Datei „Apfel.*“ und stelle den Apfel mit Hilfe des magnetischen Lasso-Werkzeugs frei!

Zauberstab

Beim Klick mit dem Zauberstab im Bild merkt sich Photoshop die Farbe des angeklickten Pixels. Die Farbe aller an den Bildpunkt angrenzenden Pixel wird mit der gemerkten Farbe verglichen. Ähneln sich die Farben, so werden die jeweiligen Pixel zur Auswahl hinzugezogen. Die Toleranz für diesen Farbvergleich ist über die Werkzeugoptionen einstellbar. Je größer die Toleranz, desto mehr Pixel werden entfernt.

Aufgabe 1

- Öffne die Datei „Zürich.psd“ und entferne den Hintergrund (Himmel) mit Hilfe des Zauberstabs!

Aufgabe 2

- Hol die von Moodle die Dateien „Winterlandschaft“ und „Rentier“!
- Versuche mit Hilfe des Zauberstabs das folgende Bild nachzugestalten.
- nützliche Hilfsmittel: Auswahl - Auswahl umkehren!



Kombinieren von Auswahlwerkzeugen

Da jedes Auswahlwerkzeug seine Vorzüge und Einschränkungen hat, ist es oft nützlich mehrere Werkzeug miteinander zu kombinieren. Die Vorgehensweise wird anhand eines Beispiels erläutert.

Aufgabe

Öffne die Datei „Lampe“! Das Ziel der Aufgabe ist es, den Lampenschirm freizustellen!

1. **Wahl des Werkzeugs:** Der Lampenschirm hat eine ähnliche Farbe wie der Hintergrund. Aufgrund des mangelnden Farbunterschieds ist an dieser Stelle das magnetische Lasso ungeeignet. Die einfache Form jedoch spricht für den Einsatz des **Polygon-Lassos**. Der Sockel der Lampe enthält sehr viele Rundungen, weshalb an dieser Stelle, das Polygon-Lasso ungeeignet ist und sich an dieser Stelle das **magnetische Lasso** deutlich besser eignet.
2. **Auswählen des Lampenschirms** mit Hilfe des Polygon-Lassos! Achtung bei der Oberkante des Schirms! Hier müssen aufgrund der leichten Rundung mehrere Punkte gesetzt werden.
3. Aktiviere in der Werkzeug-Palette das magnetische Lasso. Bevor du mit der Auswahl beginnst, muss in der Werkzeugoption der Auswahlmodus auf **Der Auswahl hinzufügen** gestellt werden (ICON-links). Der Kantenkontrast des magnetischen Lassos soll auf 7 px gesetzt werden.
4. **Auswählen des Sockels:** Bevor du mit der Arbeit beginnst, vergrößere die Ansicht auf ca. 200 %. Fahre nun mit dem magnetischen Lasso den Sockel entlang.
5. Nach dem Schließen der Sockelauswahl werden automatisch beide Auswahlen addiert. Nun sollte der Lampenschirm vollständig ausgewählt sein.
6. Wähle anschließend in der Menüleiste den Punkt **Filter - Kunstfilter** und teste verschiedene Optionen an deiner Auswahl.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:04:04_03



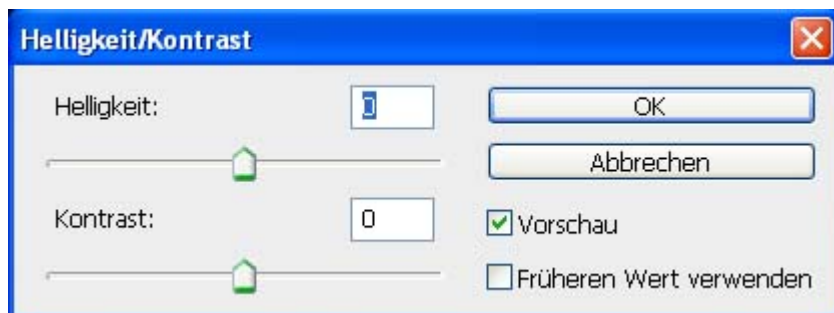
Last update: **2019/05/10 14:08**

Bildqualität optimieren

Helligkeit und Kontrast

Mittels dieser Funktion kann die Balance zwischen Lichtpartien und Schattenbereichen angepasst werden, um Bilder heller oder dunkler zu gestalten. Eine Erhöhung der Helligkeit und des Kontrastes führt in der Regel zu einer deutlichen Verbesserung der Bildqualität.

Bild - Anpassen - Helligkeit/Kontrast



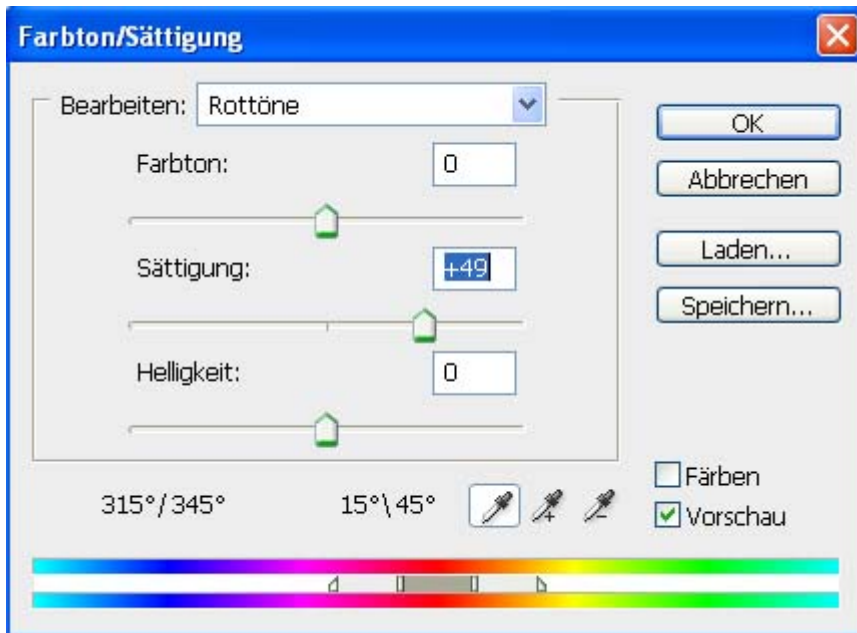
Aufgabe

Öffne das Bild „Schönbrunn.jpg“ und versuche mit Hilfe dieser Option die Qualität des Bildes zu optimieren.

Farbton und Sättigung

Mit dem Menüpunkt Bild - Anpassen - Farbton/Sättigung können die Farbtöne eines Bildes ersetzt bzw. deren Sättigung verringert und erhöht werden.

Grundsätzlich wirken sich die Änderungen auf das gesamte Bild aus. Möchte man jedoch nur gezielt einzelne Bereiche bearbeiten, so muss mit Hilfe eines geeigneten Werkzeuges zuvor eine Auswahlbereich festgelegt werden.



Wählt man die Checkbox Färben, so kann ein Bild komplett in eine neue Farbe getaucht werden.

Besonders hilfreich ist auch die Möglichkeit, einen Farbtonbereich zu definieren, auf welchen sich die Farbveränderung beschränken soll. Dazu muss im Listenfeld der entsprechende Farbbereich ausgewählt werden.

Aufgabe

Öffne das Bild „Hundertwasser1.jpg“.

Dieses Bild wurde an einem bewölkten Tag erstellt, weshalb die Farben wenig gesättigt erscheinen.

Wähle im Listenfeld die Rottöne aus und verändere die Sättigung mit Hilfe des Schieberegler.

Anschließend verbessere auch die Rot-, Gelb- und Blautöne des Bildes!

Staub und Kratzer aus einem Bild entfernen

Störungsfilter

Mit dem Hilfe der Option Filter - Störungsfilter (Rauschfilter) - Staub und Kratzer können Fehler in einem Bild in beschränktem Maße reduziert werden.

Mit dem Regler *Radius* legt man fest, wie weit das Programm nach abweichenden Pixeln suchen soll.

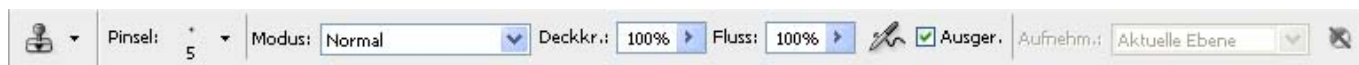
Mit dem Regler *Schwellenwert* legt man fest, wie unterschiedlich die Werte der Pixel sein müssen, damit sie beseitigt oder verändert werden.

Kopierstempel

Das Kopierstempelwerkzeug aus der Werkzeugpalette ermöglicht das Ersetzen von Bildbereichen durch andere Bereiche. Deshalb muss zuerst ein Referenzpunkt an der Stelle im Bild gesetzt werden, von der eine Kopie erzeugt werden soll. Der Referenzpunkt wird durch einen Kreis mit einem Kreuz

dargestellt. Der Referenzpunkt wird mit gedrückter ALT - Taste und einem Mausklick an die gewünschte Stelle festgelegt.

Bevor jedoch mit der Arbeit begonnen wird, ist es notwendig, in der Optionsleiste entsprechende Einstellungen vorzunehmen.



Neben den verschiedensten Optionen sollte hier auch eine geeignete Stärke der Pinselspitze ausgewählt werden.

Bei der Arbeit mit diesem Werkzeug sollte auf alle Fälle die Ansicht des Bildes stark vergrößert werden.

Aufgabe

Öffne die Datei „Kratzer_kind“ und verbessere die Qualität des sehr alten Bildes mit Hilfe des Kopierstempels. Entferne insbesondere die größeren Kratzer.

Entfernen roter Augen

- Öffne die Datei „Auge.jpg“
- Wähle mit Hilfe eines Auswahlwerkzeugs (z.B. Zauberstab) den gesamte rote Stelle aus!
- Wähle Bild - Anpassungen - Selektive Farbkorrektur: Farben „Rottöne“, Methode „Absolut“



- Setze „Schwarz“ auf 100 %!
- Um die restlichen Rotstiche zu entfernen wähle anschließend Bild - Anpassungen - Farbbalance und schiebe den obersten Farbgler richtung Cyan.



Diese Methode hat den Vorteil, dass Blitzreflexe im Auge erhalten bleiben und dadurch natürlich wirken.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:04:04_04



Last update: **2019/05/10 14:23**

5) Betriebssysteme

Ein Betriebssystem, auch **OS (operating system)** genannt, ist eine **Zusammenstellung von Computerprogrammen**, die die **Systemressourcen eines Computers wie Arbeitsspeicher, Festplatten, Ein- und Ausgabegeräte verwaltet** und diese **Anwendungsprogrammen zur Verfügung stellt**. Das Betriebssystem bildet dadurch die **Schnittstelle zwischen den Hardware-Komponenten und der Anwendungssoftware des Benutzers**. Betriebssysteme bestehen in der Regel aus einem **Kernel (deutsch: Kern)**, der die **Hardware des Computers verwaltet**, sowie speziellen **Programmen, die beim Start unterschiedliche Aufgaben** übernehmen. Zu diesen Aufgaben gehört unter anderem das Laden von **Gerätetreibern**. Betriebssysteme finden sich in fast allen Arten von Computern: Als Echtzeitbetriebssysteme auf Prozessrechnern und Eingebetteten Systemen, auf Personal Computern, Tabletcomputern, Smartphones und auf größeren Mehrprozessorsystemen wie z. B. Servern und Großrechnern.

- [5.1\) Einführung](#)
- [5.2\) Zusammenhang Hardware - Betriebssystem](#)
- [5.3\) Aufgaben, Ziele und Eigenschaften von Betriebssystemen](#)
- [5.4\) Betriebssystemarten](#)
- [5.5\) Entwicklung des Betriebssystems](#)
- [5.6\) Formatierung von Datenträgern](#)
- [5.7\) Planung eines Systems](#)
- [5.8\) Übersicht über wichtige Betriebssysteme](#)
- [5.9\) Windows](#)
- [5.10\) Linux](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05

Last update: **2019/05/14 16:02**



5.1) Einführung Betriebssysteme

Wiederholung Hardware

Wortherkunft

„Hardware“ kommt ursprünglich aus dem Englischen und bedeutet übersetzt Eisenwaren.

Hardware vs. Software

- Hardware = sind alle greifbaren/sichtbaren Elemente eines PCs
- Software = Programme und Daten, also nicht greifbare Elemente eines PCs

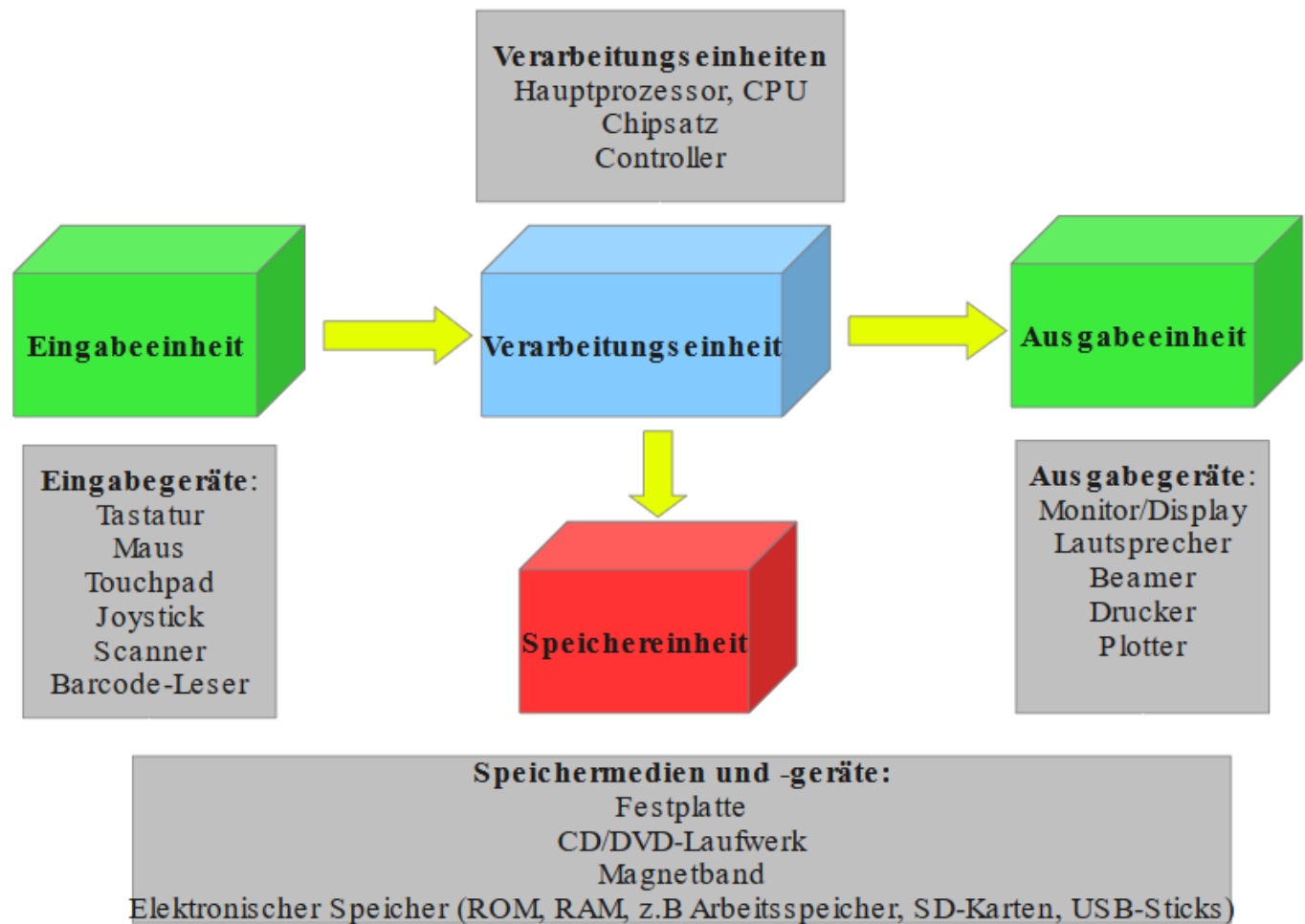
Komponenten

- **Grundbestandteile**
 - Motherboard/Mainboard - Hauptplatine
 - Central Processing Unit (CPU) - zentrale Recheneinheit (Prozessor)
 - Random Access Memory (RAM) - Arbeitsspeicher
- **Massenspeicher**
 - Festplatte (SSD, SATA)
 - Laufwerke (CD, DVD, Band,...)
- **Erweiterungskarten** (optional)
 - Grafikkarte
 - Soundkarte
 - Netzwerkkarte
- **Peripheriegeräte**
 - Eingabegeräte
 - Maus & Tastatur
 - Scanner
 - Ausgabegeräte
 - Bildschirm
 - Drucker

EVA/IPO Prinzip

Das **EVA**-Prinzip ist ein Grundprinzip für die Datenverarbeitung und beschreibt die Reihenfolge in der Daten verarbeitet werden.

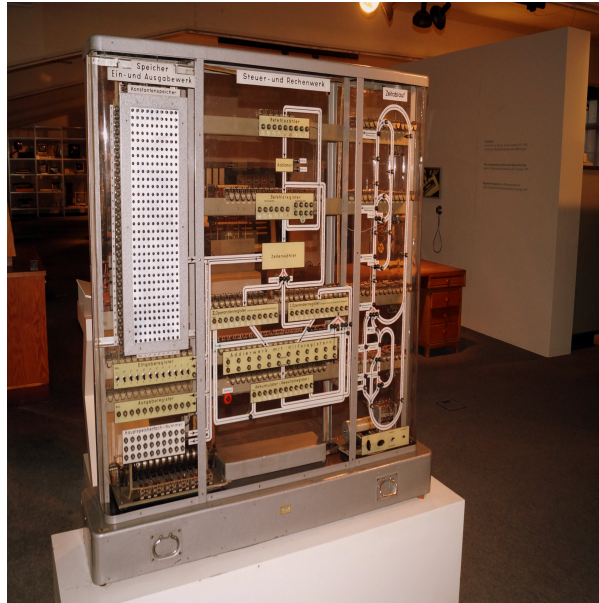
Eingabe (**I**ntput) - **V**erarbeitung (**P**rocess) - **A**usgabe (**O**utput)



EVA Prinzip

Von-Neumann-Architektur (VNA)

Die **Von-Neumann-Architektur** ist ein Modell für Computer und bietet die Grundlage für alle heutigen Computer. Das Modell wurde von [Johann von Neumann](#) im Jahr 1945 entwickelt. Heute ist Johann von Neumann unter seinem amerikanischen Namen John von Neumann bekannt.

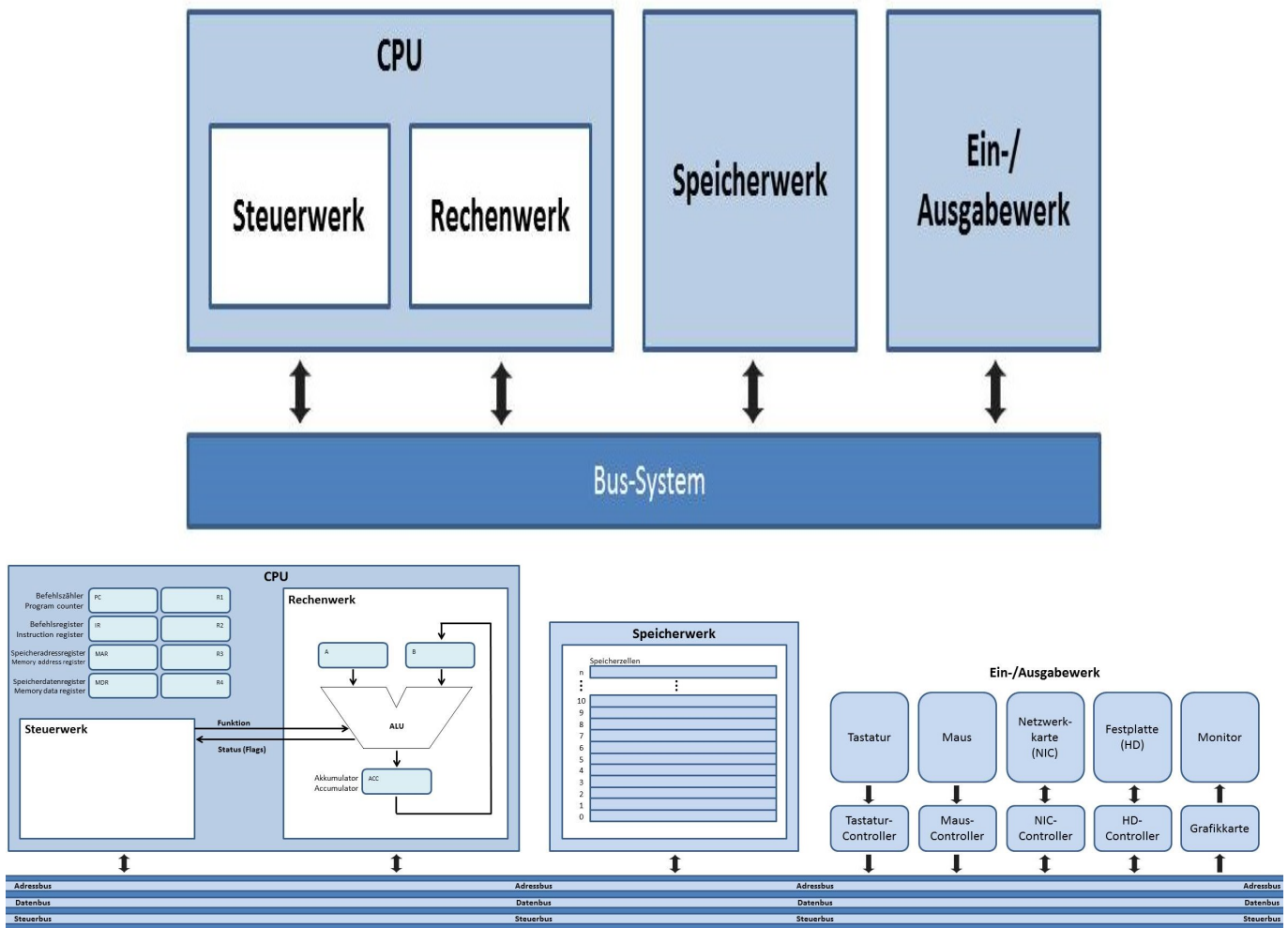


Er revolutionierte die bisherigen Computer, da durch seine Architektur verschiedene Programme auf derselben Hardware laufen konnten. Seine Struktur hat es erstmals ermöglicht, dass mehrere Probleme von ein und derselben Hardware gelöst werden können.

Komponenten

Ein Von-Neumann-Rechner beruht auf folgenden Komponenten, die bis heute in Computern verwendet werden:

- **ALU** (Arithmetic Logic Unit) – Rechenwerk, selten auch Zentraleinheit oder Prozessor genannt, führt Rechenoperationen und logische Verknüpfungen durch. (Die Begriffe Zentraleinheit und Prozessor werden im Allgemeinen in anderer Bedeutung verwendet.)
- **Control Unit – Steuerwerk oder Leitwerk**, interpretiert die Anweisungen eines Programms und verschaltet dementsprechend Datenquelle, -senke und notwendige ALU-Komponenten; das Steuerwerk regelt auch die Befehlsabfolge.
- **Bussystem**: Dient zur Kommunikation zwischen den einzelnen Komponenten (Steuerbus, Adressbus, Datenbus)
- **Memory – Speicherwerk** speichert sowohl Programme als auch Daten, welche für das Rechenwerk zugänglich sind.
- **I/O Unit – Eingabe-/Ausgabewerk** steuert die Ein- und Ausgabe von Daten, zum Anwender (Tastatur, Bildschirm) oder zu anderen Systemen (Schnittstellen).



Register

- **Befehlszähler (Program Counter - PC)**

Enthält die Speicheradresse vom Speicherwerk des aktuellen Befehls. (Startadresse = 0x0000). Wird nach jedem Befehl um 1 erhöht.

- **Befehlsregister (Instruction Register - IR)**

Speichert den vom Speicherwerk zurückbekommenen Befehl.

- **Speicheradressregister (Memory Address Register - MAR)**

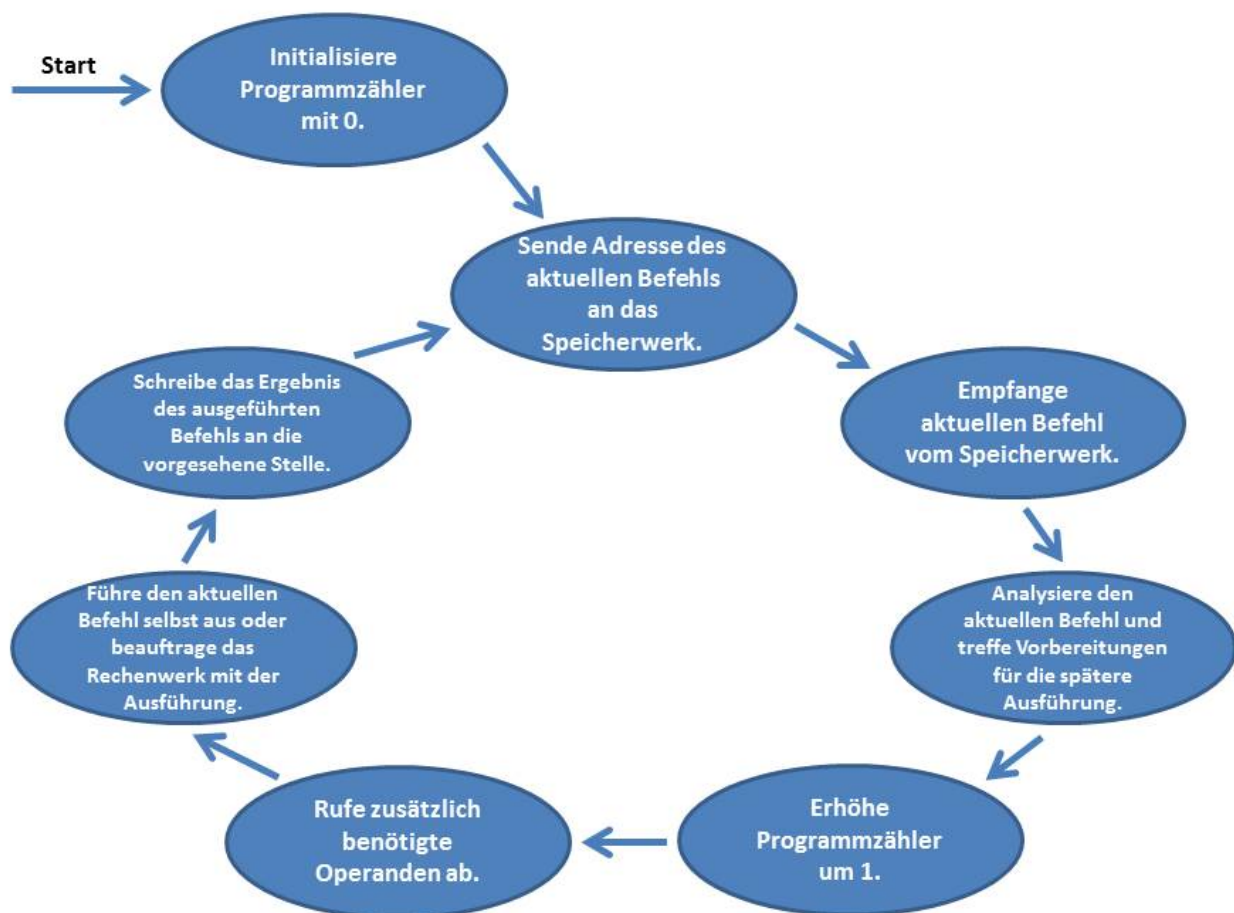
Ausschließlich für die Kommunikation zwischen Steuerwerk und Rechenwerk. Im MAR legt das Steuerwerk jeweils die Adresse ab, welche im Speicherwerk angesprochen werden soll.

- **Speicherdatenregister (Memory Data Register - MDR)**

Ausschließlich für die Kommunikation zwischen Steuerwerk und Rechenwerk. Bei einem Lesezugriff auf die Speicherzelle wird der vom Speicherwerk über den Datenbus bereitgestellte Wert im MDR abgelegt und kann von hier aus weiter verarbeitet werden. Bei einem Schreibzugriff muss sich im MDR der zu schreibende Wert befinden, so dass er über den Datenbus an das Speicherwerk übermittelt werden kann.

Prozesszyklus

1. Initialisiere das Befehlszählerregister (Program Counter- PC) mit 0 (Start)
2. Sende Adresse des aktuellen Befehls zum Speicherwerk
3. Empfange aktuellen Befehl vom Speicherwerk und speichere diesen in das Befehlsregister (Instruction Register - IR)
4. Analysiere aktuellen Befehl und treffe Vorbereitungen für die spätere Ausführung (Welcher Befehl und was ist dazu notwendig?)
5. Erhöhe den Befehlszähler (PC) um 1
6. Rufe zusätzlich benötigte Operanden ab (z.B.: Befehl ADD OP1 OP2)
7. Führe den Befehl selbst (Steuerwerk) aus oder beauftrage das Rechenwerk für die Ausführung
8. Schreibe das Ergebnis des ausgeführten Befehls an die vorgesehene Stelle



Arbeitsweise des Steuer- und

Rechenwerks



Arbeitsweise des Speicherwerks



Befehlszähler und Befehlsregister im Zusammenspiel mit dem Bus- System



Arbeitsweise des Steuerwerks

Die 7 Prinzipien der Von-Neumann-Architektur

- Rechner besteht aus fünf Funktionseinheiten
- Struktur des Rechners ist unabhängig vom zu bearbeitenden Problem. Zur Lösung eines Problems muss Programm im Speicher abgelegt werden.
- Programme, Daten und Ergebnisse werden im selben Speicher abgelegt.
- Der Speicher ist in fortlaufenden nummerierten Zellen unterteilt. Über die Adresse einer Speicherzelle kann auf den Inhalt zugegriffen werden.
- Aufeinanderfolgende Befehle eines Programms werden in aufeinanderfolgende Speicherzellen abgelegt.
- Durch Sprungbefehle kann von der gespeicherten Befehlsreihenfolge abgewichen werden.
- Es gibt zumindest
 - arithmetische Befehle (Addition, Subtraktion, Multiplikation)
 - logische Befehle (EQUAL, NOR, AND, OR)
 - Transportbefehle, z.B. von Speicher zu Rechenwerk und für Ein- und Ausgabe
- Alle Daten (Befehle, Adressen, usw.) werden binär kodiert

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_01



Last update: **2019/05/14 15:43**

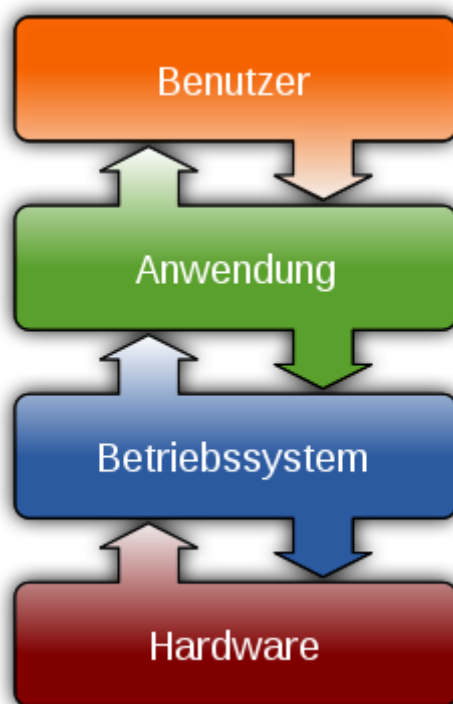
5.2) Zusammenhang Hardware - Betriebssysteme

Grundlegendes

Ein Rechner besteht zunächst aus den sichtbaren und greifbaren Komponenten der Hardware. Damit aber mit dem Rechner auch gearbeitet werden kann, benötigt er noch geeignete Programme. Nach dem Von-Neumannschen Prinzipien ist ein Rechner als Universalrechner konstruiert. Das heißt die Erledigung unterschiedlicher Aufgaben wird ausschließlich über Programme (auch Software genannt) gesteuert.

Da sich aber nicht jeder Softwareentwickler um die Details zur Verwaltung der einzelnen Hardwarekomponenten (z.B.: Steuerungsinformationen für Plattenspeicher, Speicherverwaltung, Prozessverwaltung etc.) kümmern sollte, erscheint es sinnvoll diese Komponenten zentral und damit allgemein verfügbar bereitzustellen. Andernfalls würde jeder Entwickler viel Zeit verschwenden.

Somit entwickelte man eine Ebene (=Betriebssystemebene) auf der verschiedenste Verwaltungsprogramme bereits laufen und allen Anwendern bzw. Anwendungsprogrammen eine vereinfachte Sicht auf die Hardware zur Verfügung stellen.



Folgendes gilt, dass das Betriebssystem quasi der Mittler (=Übersetzer) zwischen Hardware und Benutzer ist. Dadurch kann die Komplexität der darunterliegenden Systemarchitektur verborgen werden und dem Anwender wird eine einfache und verständliche Schnittstelle (Interface) angeboten. Er kann sich somit voll und ganz auf die Implementierung seiner Funktion konzentrieren und muss sich nicht mit komplexen Maschinenbefehlen herumschlagen.

Des Weiteren verwaltet das Betriebssystem die Ressourcen des Rechners, also alle physikalischen

Geräte (Prozessoren, Speicher, Grafikkarte,...). Es teilt also die Ressourcen des Gesamtsystems gerecht an die konkurrierenden Anwenderprogramme auf. Ohne einer solchen Aufteilung wäre eine sinnvolle Benützung des PCs nicht möglich.

Als normaler Computeranwender befinden wir uns auf der **Benutzerebene**, wo der Rechner als Arbeitsgerät auch für technische Laien dienen muss.

Zwischen dem vom Anwender intuitiv zu bedienenden Rechner und den Fähigkeiten der Hardware klafft eine riesige Lücke, die vom

- Betriebssystem (Datei-, Prozess- und Speicherverwaltung) und dem
- grafischen Bediensystem (Menüs, Fenster, Maus)

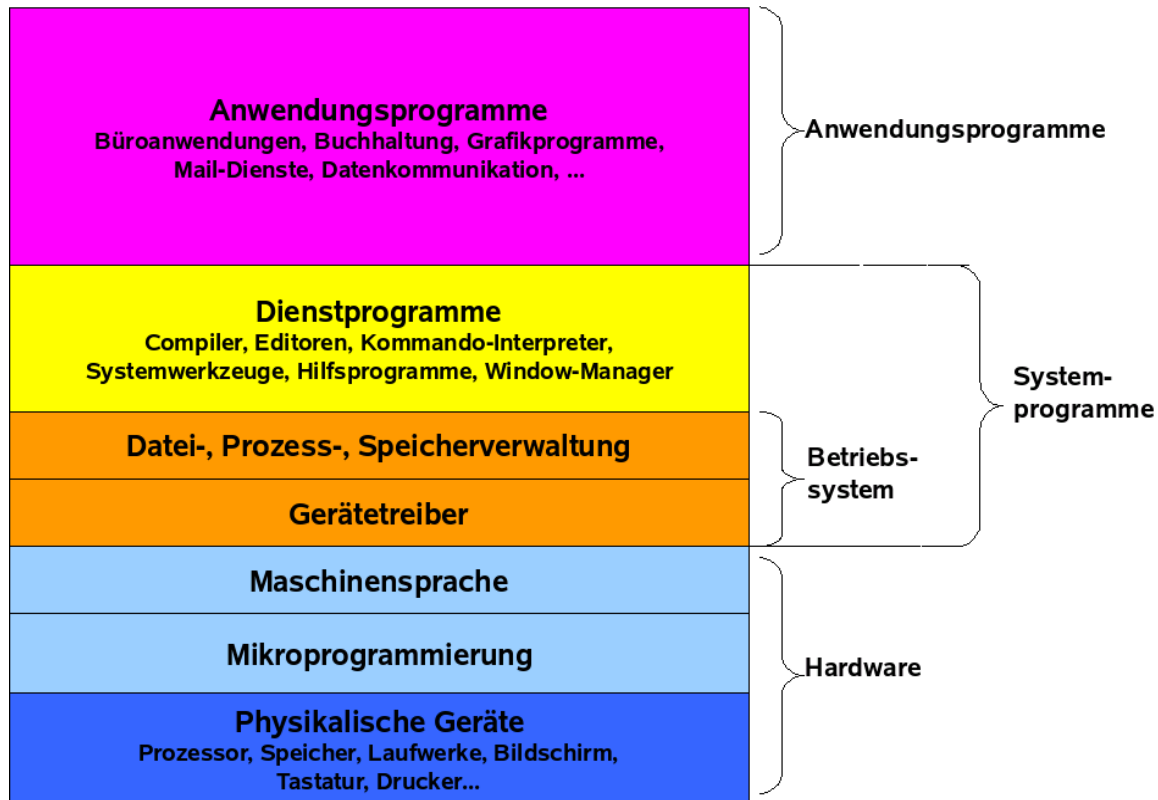
überbrückt wird.



Jede Schicht fordert von der niedrigeren Schicht Dienste an. Diese wiederum benötigt zur Erfüllung der Anforderungen selber Dienste von der nächsten Schicht. Auf diese Weise setzen sich die Anforderungen in die tiefen Schichten fort, bis die Hardware zu geeigneten Aktionen veranlasst wird.

Anders betrachtet, bietet jede Schicht der jeweils höherliegenden Schicht Dienste an.

Schichten eines Computersystems



Physikalische Geräte

Die unterste Schicht enthält die physikalischen Geräte. Sie besteht aus integrierten Schaltungen, Drähten, Stromversorgung usw.

Mikroprogrammierung/Microcode

Aufgabe des Microcodes ist die direkte Steuerung der physikalischen Geräte. In Microcode entworfene Programme werden von einem Interpreter in entsprechende Steueranweisungen übersetzt und an die Geräte übermittelt. Die Instruktionen der Maschinensprache (z.B.: ADD, MOVE, JUMP,...) werden in einzelne kleine Microcodeprogramme übersetzt.

Maschinensprache

Die Menge von Instruktionen die das Mikroprogramm ausführen kann wird als Maschinensprache bezeichnet. Maschinensprache besteht aus zahlreichen elementaren Maschinenbefehlen die im Prozessor in Microcode zu einem Programm übersetzt und ausgeführt werden. Wichtig ist, dass jeder Maschinenbefehl durch ein festgelegtes Mikroprogramm implementiert bzw. fest verdrahtet ist.

Da die Maschinensprache hardwarespezifisch ist, wird sie noch der Hardware zugeordnet.

Betriebssystem

Das Betriebssystem vermindert die Komplexität der Hardware und deren Verwaltung um dem Programmierer eine angemessene Menge an Instruktionen zur Verfügung zu stellen, mit denen er effizienter arbeiten kann.

Systemprogramme / Dienstprogramme

Das sind vom Betriebssystemhersteller mitgelieferte Programme wie z.B.: Window-Manager, Compiler für das Kompilieren von C-Programmen, Editoren, Systemwerkzeuge wie z.B.: Taskmanager, Explorer ...

Anwenderprogramme

Die oberste Schicht stellt den Benutzern des Systems Anwendersoftware zur Verfügung. Diese Programme setzen entweder auf den Dienstprogrammen oder unmittelbar auf dem Betriebssystem auf. Beispiele dafür sind:

- Browser
- Microsoft Office
-

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_02

Last update: **2019/05/14 15:43**



5.3) Aufgaben, Ziele & Eigenschaften von Betriebssystemen

Hauptaufgaben

1) Abstraktion (Vereinfachung) der Hardware --> abstrakte Programmierschnittstelle

2) Verwaltung von Systemressourcen (CPU, Speicher, Netzwerk, Drucker, Platten..)

- **Prozessverwaltung**
- **Speicherverwaltung**
- **Dateiverwaltung**

Der Rechner mit seinen Peripheriegeräten stellt eine Fülle von Ressourcen zu Verfügung, wie CPU, Hauptspeicher, Plattenspeicher, externe Geräte. Die Verwaltung dieser Ressourcen ist Aufgabe des Betriebssystems.

Der Aufruf eines Programms führt oft zu vielen gleichzeitig und unabhängig voneinander ablaufenden Teilprogrammen, auch **Prozesse** genannt.

Ein Prozess ist also ein eigenständiges Programm mit eigenem Speicherbereich, der vor dem Zugriff durch andere Prozesse geschützt ist. Allerdings ist es erlaubt, dass verschiedene Prozesse untereinander *kommunizieren*, also Daten untereinander austauschen.

Dabei hat das Betriebssystem die Aufgabe, die *Kommunikation* zwischen diesen Prozessen zu verwalten, damit sich die gleichzeitig aktiven Prozesse nicht untereinander beeinträchtigen oder gar zerstören. Das Betriebssystem verwaltet unter anderem also gleichzeitig aktive Prozesse, so dass einerseits keiner benachteiligt wird, andererseits aber kritische Prozesse mit Priorität behandelt werden.

Selbstverständlich können Prozesse aber nicht wirklich gleichzeitig ablaufen. Das Betriebssystem erzeugt aber eine scheinbare Parallelität dadurch, dass jeder Prozess immer wieder eine kurze Zeitspanne (wenige Millisekunden) an die Reihe kommt, dann unterbrochen wird, während andere Prozesse bedient werden. Nach kurzer Zeit ist der unterbrochene Prozess wieder an der Reihe und setzt seine Arbeit fort. Diesen Vorgang nennt man **Multitasking**.

Ähnlich verhält es sich mit der Verwaltung des Hauptspeichers, in dem nicht nur der Programmcode, sondern auch die Daten der vielen Prozesse gespeichert werden. Neuen Prozessen muss freier Hauptspeicher zugeteilt werden, der Speicher beendeter Prozesse muss wiederverwendet werden.

Die dritte wichtige Aufgabe ist die **Dateiverwaltung**. Damit ein Benutzer sich nicht darum kümmern muss, in welchen Sektoren auf welchen Spuren noch Platz ist um den gerade geschriebenen Text zu speichern, stellt das Betriebssystem das Konzept „Datei“ als Behälter für Daten aller Art zur Verfügung. Die Übersetzung von Dateien und ihren Namen in bestimmte Spuren, Sektoren und Köpfe der Festplatte nimmt das *Dateisystem* als Bestandteil des Betriebssystems vor.

Eigenschaften von Betriebssystemen

- Betriebssystem muss **änderbar** sein (durch neue Hardware, Software bzw. bei Sicherheitslücken)
- **modular** und **klar strukturiert** aufgebaut
- gut **dokumentiert** -> Schnittstellenbeschreibung!!

Ziele von Betriebssystemen

- **Bequemlichkeit** - aus Sicht des Benutzers
- **Effizienz** - Nutzung der Ressourcen
- **Entwicklungsfähigkeit** - Neue Dienste

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_03



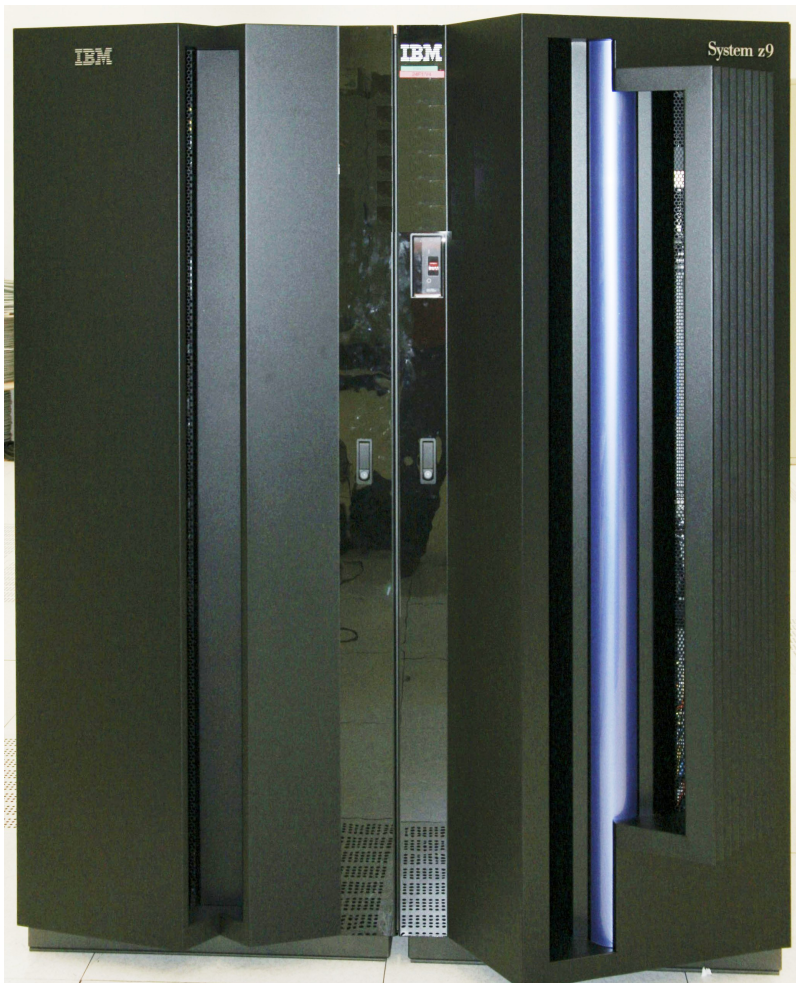
Last update: **2019/05/14 15:44**

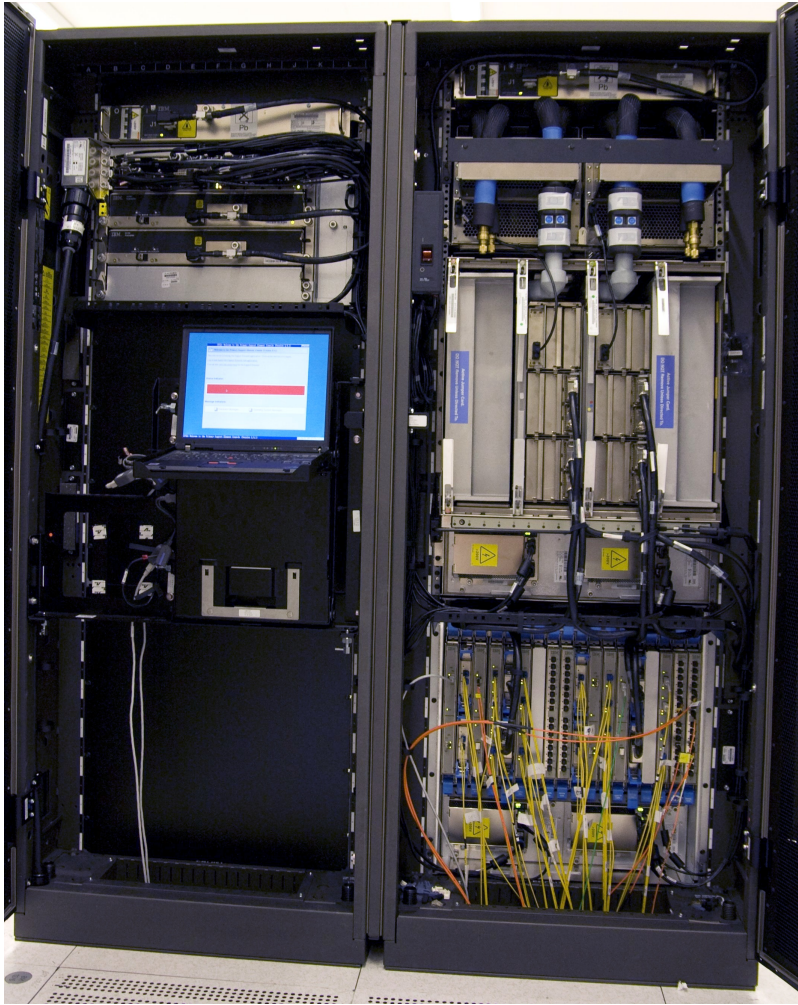
5.4) Betriebssystemarten

Man unterscheidet mehrere Arten von Betriebssystemen, die im Laufe der Zeit entstanden sind.

Mainframe-Betriebssysteme

- Betriebssysteme für Großrechner
- Einsatz: Webserver, Datenbankserver, E-Commerce, Business-to-Business (B2B)
- Viele Prozesse gleichzeitig mit hohem Bedarf an schneller Eingabe/Ausgabe
- Beispiele sind MVS, OS/390, z/OS 2.1 -> zumeist IBM-Großrechner





Server-Betriebssysteme

- Betriebssysteme die auf Servern laufen
- Bieten verschiedene Dienste im Netzwerk an (Dateidienste, Webdienste,...)
- Dienste stehen vielen Benutzern im Netz zur Verfügung z.B.: Netzlaufwerk
- Aktuelle Beispiele sind Windows Server 2016 bzw. Red Hat Enterprise Linux 7.2

PC(Desktop)-Betriebssysteme

- Betriebssystem für Personalcomputer
- Meist nur 1 oder wenige Benutzer (über Netzwerk)
- Einsatz: Programmierung, Textverarbeitung, Spiele, Internetzugriff
- Mehrere Programme pro Benutzer → quasi-parallel
- Aufteilung der Prozesse auf vorhandene Kerne
- Zuteilung der Systemressourcen
- Beispiele: Linux, Windows, Mac OS X

Echtzeit-Betriebssysteme

- Einhaltung von harten Zeitbedingungen

- Einsatz: Steuerung von maschinellen Fertigungsanlagen, Steuerung von Ampeln, Robotorsteuerung...
- Aktion in einem fest vorgegebenen Zeitintervall

Eingebettete Systeme / Embedded Systems

- = Computer die man nicht unbedingt gleich sieht
- Einsatz: Fernseher, Handy, Auto,...
- Meist Echtzeitanforderungen
- Wenig Ressourcen zur Verfügung (geringer Stromverbrauch, wenig Arbeitsspeicher)
- Beispiele: iOS, Android, Windows Phone

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_04



Last update: **2019/05/14 15:45**

5.5) Historische Entwicklung von Betriebssystemen

Arbeitsauftrag

Recherchiere im Internet hinsichtlich den historischen Entwicklungen von Betriebssystemen.

Beschreibe und erkläre die folgenden Entwicklungsstadien (= 4 Generationen) und gib die ungefähre Zeitspanne an:

1. Generation - Serielle Systeme
2. Generation - Einfache Stapelverarbeitungssysteme
3. Generation - ICS, Multiprogramming, Timesharing
4. Generation - Personal Computer

Das Ergebnis soll ein Word-Dokument (gegliedert in den 4 Generationen) im Umfang von 2-4 Seiten exklusive Titelblatt (Download unter

Titelblatt

- vor der Abgabe bitte den Namen im Titelblatt und den Dateinamen auf Betriebssysteme_NACHNAME.docx ändern) sein (d.h. mindestens 3 Seiten).

Letzter Abgabetermin ist der 21.02.2019 @ 23:59.

Die bis dahin abgegebenen Arbeitsaufträge werden bewertet und zählen zur Teilnote Praktische Arbeiten (PA)!!

Nicht abgegebene Arbeitsaufträge werden negativ beurteilt.

Musterlösung unter

Entwicklung Betriebssysteme

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_05

Last update: **2019/05/14 15:48**



5.6) Formatierung von Datenträgern

5.6.1) Stufen der Formatierung

Wenn man von Formatierung spricht, unterscheidet man dabei mehrere Stufen:

1. **Low-Level-Formatierung** = physikalische Einteilung eines Speichermediums
2. **Partitionierung** = logische Einteilung des Speichermediums in zusammenhängende Strukturen
3. **High-Level-Formatierung** = die logische Einteilung der Partitionsstruktur mit einem Dateisystem durch eine Software (meist durch das Betriebssystem).

Soweit durch die Beschaffenheit, industrielle Standards oder durch spezielle Verwendung die physikalische Einteilung des Mediums als Norm feststeht, ist eine separate Low-Level-Formatierung nicht erforderlich. In diesen Fällen können beide Vorgänge gleichzeitig vorgenommen werden, so zum Beispiel bei Disketten, CD-ROM, CD-RW oder DVD-ROM/RW.

5.6.1.1) Physikalische Formatierung (Low-Level-Formatierung)

Damit eine Festplatte logisch formatiert werden kann, muss sie physikalisch formatiert sein.

Die physikalische Formatierung eines Festplattenlaufwerks (auch als Zwei-Stufen-Formatierung bezeichnet) wird in der Regel vom Hersteller durchgeführt.

Durch die physikalische Formatierung wird eine Festplatte in ihre physikalischen Grundbausteine unterteilt: **Spuren**, **Sektoren** und **Zylinder**. Durch diese Elemente wird die Art und Weise vorgegeben, mit der Daten physikalisch auf der Festplatte aufgezeichnet und von dort gelesen werden können.

Spuren

Die **Spuren** sind konzentrische Kreispfade, die auf jede Scheibenseite geschrieben werden, wie auf einer Schallplatte oder einer CD. Die Spuren werden mit einer Nummer identifiziert, die mit Spur 0 am äußeren Rand einsetzt.

Zylinder

Der Spurensatz, der auf allen Seiten der Platte im gleichen Abstand von der Mitte angelegt ist, wird als **“Zylinder”** bezeichnet. Hardware und Software arbeiten häufig mit diesen Zylindern.

Sektoren

Die Spuren sind in Bereiche, sogenannte **“Sektoren”** oder **„Blöcke”** unterteilt, in denen eine festgeschriebene Datenmenge gespeichert wird. Die Sektoren werden in der Regel für eine Speicherkapazität von 512 Byte (ein Byte besteht aus 8 Bit) formatiert.

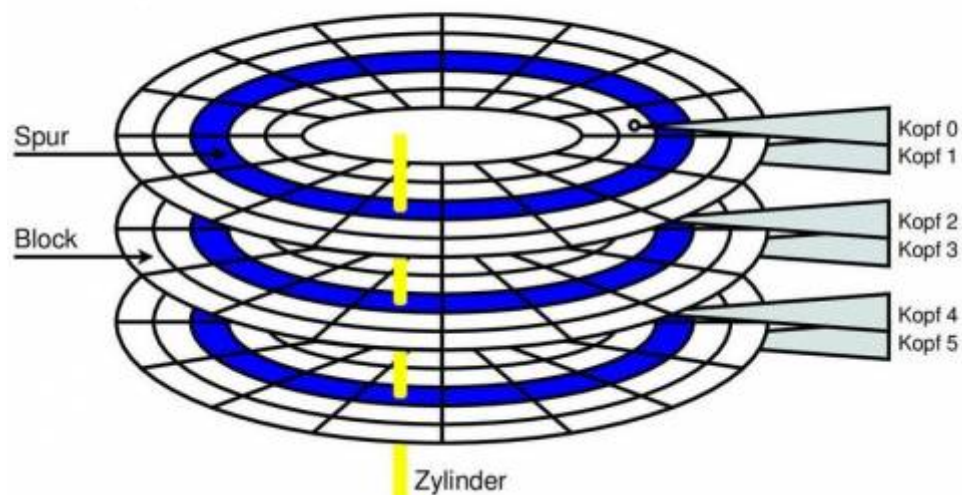
Fehlerhafte Sektoren

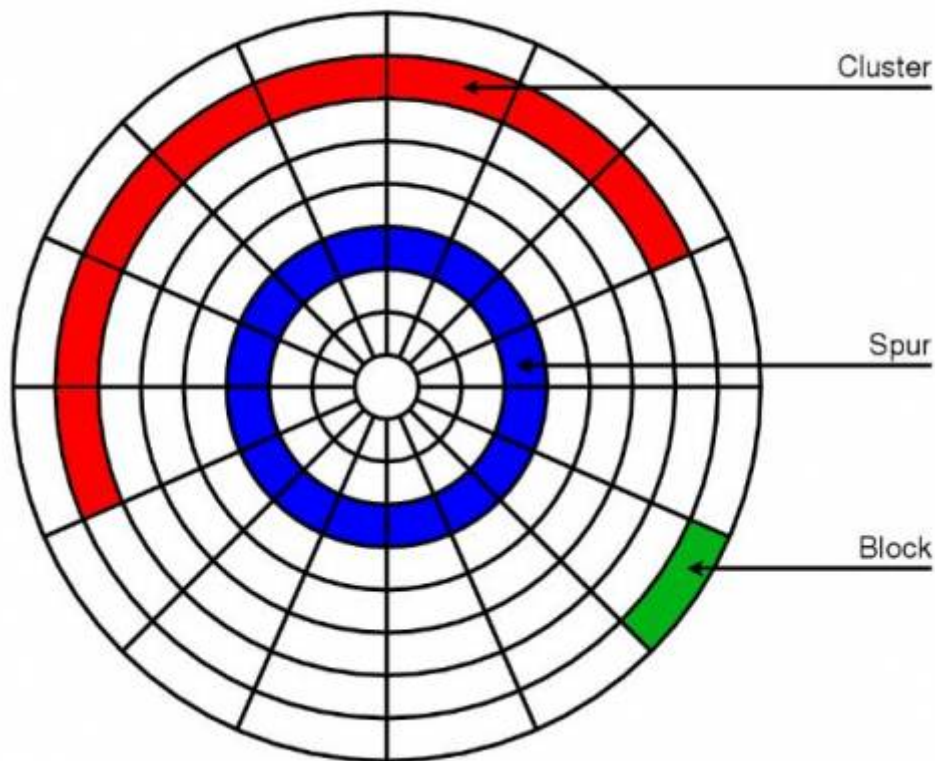
Nachdem ein Festplattenlaufwerk physikalisch formatiert wurde, kommt es gelegentlich vor, dass die magnetischen Eigenschaften der Eisenoxydschicht an einigen Stellen auf der Festplatte nach und nach an Wirksamkeit verlieren. Dies hat zur Folge, dass es für die Schreib-/Leseköpfe der Festplatte in zunehmenden Maße schwieriger wird, ein Bit-Muster auf der Festplatte zu speichern, das später von der Festplatte gelesen werden kann.

Wenn dieser Prozess einsetzt, bezeichnet man die Sektoren, in denen Daten nicht mehr zuverlässig gespeichert werden können, als **“fehlerhafte Sektoren”**. Glücklicherweise ist die Qualität moderner Festplatten so hoch, dass solche **“fehlerhaften Sektoren”** nur sehr selten auftreten.

Darüber hinaus sind die modernen Computer normalerweise in der Lage, einen fehlerhaften Sektor zu identifizieren und ihn als solchen zu kennzeichnen, damit er nicht mehr genutzt wird; es wird dann ein alternativer Sektor benutzt.

- Alle Spuren auf allen Platten bei einer Position des Schwingarms bilden einen Zylinder





Adressierung der Daten auf einer (magnetischen) Festplatte

CHS - Adressierung

Bei Festplatten mit einer Kapazität < 8 GB werden die einzelnen Sektoren mit der sogenannten **„Zylinder-Kopf-Sektor-Adressierung (Cylinder-Head-Sector-Adressierung)“** durchgeführt.

Jeder Sektor lässt sich mit **CHS** klar lokalisieren und adressieren.

Einschränkungen durch das BIOS

- Zylinder = 10 Bits
- Kopf = 4 Bits -> später 8 Bits (erweitertes CHS)
- Sektor/Spur = 8 Bits

Maximaler Speicherplatz früher: $1.024 \text{ Zylinder} * 16 \text{ Köpfe} * 63 \text{ Sektoren/Spur} * 512 \text{ Byte/Sektor} = 528.482.304 \text{ Byte} = 504 \text{ MB}$

Maximaler Speicherplatz heute: $1.024 \text{ Zylinder} * 255 \text{ Köpfe} * 63 \text{ Sektoren/Spur} * 512 \text{ Byte/Sektor} = 8.422.686.720 \text{ Byte} = 7,844 \text{ GB}$

- Größe einer Spur berechnen?
- Größe einer Scheibe berechnen?

LBA- Adressierung

Seit der Einführung der logischen Blockadressierung (**Logical Block Addressing (LBA)**) 1995 werden alle Sektoren auf der Festplatte von 0 beginnend durchnummeriert. Durch LBA wurde das

BIOS - und CHS-Korsett umgangen und es kann damit jede derzeit gängige Festplatte voll adressiert werden. Die Zahlen zu Zylinder, Köpfen und Sektoren auf der Festplatte haben heute nichts mehr mit der tatsächlichen Lage der Sektoren auf der Festplatte zu tun. Es handelt sich um eine logische Plattengeometrie, die nur aus Kompatibilitätsgründen existiert

5.6.1.2) Partitionierung von Festplatten

Eine Partition ist ein zusammenhängender Bereich einer Festplatte, der in der Regel ein Dateisystem enthält.

Wenn man einen PC oder ein Notebook mit vorinstalliertem Windows kaufen, enthält die Festplatte zumeist zwei Partitionen: eine winzige Partition mit Windows-Boot-Dateien und eine zweite Partition, die den Rest der Festplatte füllt und Windows enthält. Unter Umständen kann es auch weitere Partitionen geben, die beispielsweise ein Recovery-System enthalten.

Um mehrere Betriebssysteme (Windows, Linux etc.) auf einem Rechner zu installieren, benötigt man mehrere Partitionen. Für jedes Betriebssystem ist mindestens eine Partition erforderlich; für Linux sind sogar mehrere Partitionen sinnvoll.

Master-Boot-Record (MBR)

Der Master-Boot-Record ist der **erste Sektor** auf der Festplatte. Er beinhaltet zum einen den „**Bootstrap**“. Dies ist ein Programm, das von dem BIOS aufgerufen wird, um das eigentliche Betriebssystem zu laden. Zum anderen enthält dieser Sektor auch eine Beschreibung, ob / wie die Festplatte in unterschiedliche Bereiche (Partitionen) unterteilt ist. Diese Beschreibung erfolgt in der sogenannten „Partitionstabelle“. Sie enthält für jede Partition einen Eintrag. Dieser besteht aus der Lage der Partition auf der Festplatte und dem „Typ“ dieser Partition.

Die Master-Boot-Record Partitionstabelle ist eine Tabelle, welche die Unterteilung der Festplatte in Partitionen beschreibt. Sie ist seit der Verbreitung von Festplatten im Jahr 1980 fest definiert und wird von allen Betriebssystemen zwingend vorausgesetzt.

Aus historischen Gründen kann diese Partitionstabelle nur vier Einträge (primäre Partitionen oder erweiterte Partitionen) aufnehmen. Das Format dieses Master-Boot-Records (Bootstrap / Partitionstabelle) ist fest definiert und wird von allen Betriebssystemen zwingend vorausgesetzt.

Partitionsarten

Es gibt im wesentlichen zwei Partitionsarten: **Primärpartitionen** und **erweiterte Partitionen**. Darüber hinaus lassen sich erweiterte Partitionen noch weiter in **logische Partitionen** unterteilen. Sie können bis zu vier Hauptpartitionen auf dem Festplattenlaufwerk definieren, wovon eine als erweiterte Partition definiert werden kann, d.h. maximal vier Primärpartitionen oder drei Primärpartitionen und eine erweiterte Partition.

a) Primärpartitionen

In einer Primärpartition können Sie jedes beliebige Betriebssystem sowie Datendateien wie z.B. Programm- und Benutzerdateien speichern. Eine Primärpartition ist logisch formatiert, damit sich ein Dateisystem darauf definieren lässt, das mit dem auf der Partition installierten Betriebssystem kompatibel ist.

Wenn Sie mehrere Betriebssysteme auf Ihrem Festplattenlaufwerk installieren müssen, ist es in der Regel notwendig, mehrere Primärpartitionen zu erstellen, da die meisten Betriebssysteme (vor allem Windows) nur von einer Primärpartition gebootet werden können. Pro Festplatte ist es möglich 4 primäre Partitionen einzurichten, ohne den Bootsektor der Festplatte anzupassen.

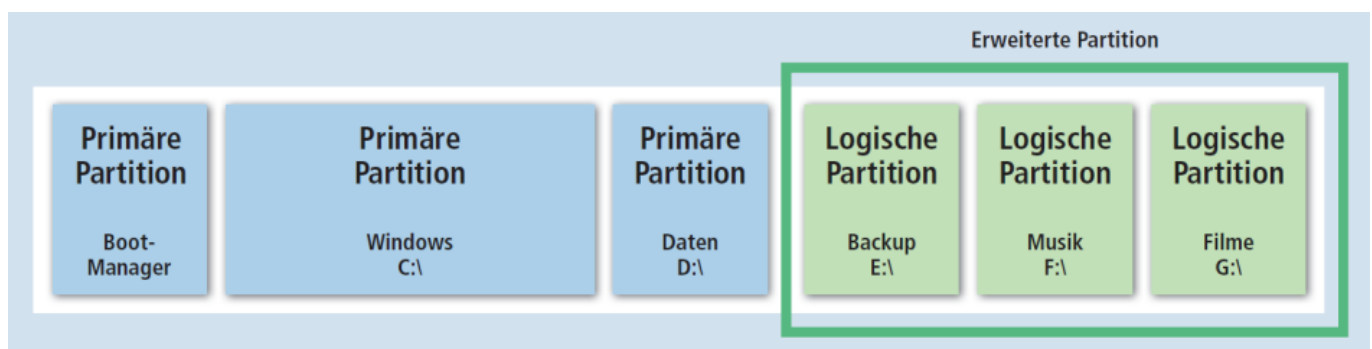
b) Erweiterte Partitionen

Die erweiterte Partition wurde entwickelt, um die willkürliche 4-Partitionen-Begrenzung zu umgehen. Es handelt sich im Prinzip um einen Bereich, in dem Sie den Datenträgerspeicher weiter physikalisch unterteilen können, indem Sie eine unbegrenzte Anzahl logischer Partitionen definieren (weitere physikalische Unterteilungen des Datenträgerspeichers).

Eine erweiterte Partition dient nur indirekt der Datenspeicherung. Der Benutzer muss in der erweiterten Partition logische Partitionen definieren, in denen die Daten gespeichert werden. Logische Partitionen müssen logisch formatiert sein; auf jeder logischen Partition kann ein anderes Dateisystem definiert sein. Nach der logischen Formatierung gilt jede logische Partition als separater Datenträger.

c) Logische Partitionen

Logische Partitionen können nur innerhalb einer erweiterten Partition definiert werden und sind nur für die Speicherung von Datendateien und Betriebssystemen vorgesehen, die von einer logischen Partition gebootet werden können (z.B. Linux und Windows NT). Betriebssysteme, die von einer logischen Partition gebootet werden können, sollten gewöhnlich in einer logischen Partition installiert werden; damit können Primärpartitionen für andere Zwecke freigehalten werden.



Datenträgerverwaltung

Datei Aktion Ansicht ?

← → ↻ ? 📁 🗑️ 📄 📂 🔍 🛠️

Volume	Layout	Typ	Dateisystem	Status	Kapazität	Freier Sp...	% frei
System-reserviert	Einfach	Basis	NTFS	Fehlerfrei (...)	500 MB	166 MB	33 %
Windows (C:)	Einfach	Basis	NTFS	Fehlerfrei (...)	21,74 GB	9,37 GB	43 %
Daten (D:)	Einfach	Basis	NTFS	Fehlerfrei (...)	9,76 GB	9,72 GB	100 %

< >

CD 0
CD (E:)
Kein Medium

Datenträger 0
Basis
32,00 GB
Online

	System-reserviert	Windows (C:)	Daten (D:)
	500 MB NTFS Fehlerfrei (System, Akt)	21,74 GB NTFS Fehlerfrei (Startpartition, Auslagerung)	9,76 GB NTFS Fehlerfrei (Primäre Partition)

■ Nicht zugeordnet ■ Primäre Partition

/dev/sda - GParted

GParted Edycja Widok Urządzenie Partycja Pomoc

Nowy Usun Zmień rozmiar/przeniesz Skopiuj Wklej Zastosuj

/dev/sda (93.16 GiB)

/dev/sda7
84.29 GiB

Partycja	System plików	Punkt montowania	Rozmiar	Wykorzystanych	Niewykorzystanych	Flagi
▼ /dev/sda1	extended		93.16 GiB	---	---	boot
/dev/sda5	ext4	/	7.91 GiB	5.82 GiB	2.09 GiB	
/dev/sda6	linux-swap		972.65 MiB	---	---	
/dev/sda7	ext4	/home	84.29 GiB	43.99 GiB	40.31 GiB	

0 zaplanowanych operacji

Frage: Wie finde ich die aktuelle Partitionierung von meinem PC?

Gründe für Partitionen

Das Problem früher waren die Festplatten-Controller die nicht in der Lage waren, einen größeren Adressbereich anzusprechen. Und, der technische Fortschritt und die höheren Kapazitäten von Festplatten wurden schneller eingeführt als neue und bessere Dateisysteme. Vor allem unter Windows-Betriebssystemen war das FAT-Dateisystem (File Allocation Table) lange führend. FAT ermöglichte durch die Zusammenführung mehrerer Blöcke zu einer logischen Ansprecheinheit (Cluster), um die Adressierungsbeschränkung zu umgehen. Es hatte den Nachteil, dass es Festplatten nur bis zu einer bestimmten Kapazität verwalten und die Dateien nicht besonders platzsparend speichern konnte.

Beschränkungen der Partitionsgröße vom Dateisystem:

- FAT16 \Rightarrow max. Partitionsgröße = 2 GByte
- Nachfolger FAT32 \Rightarrow 2 TByte
- Über 32 GByte verwendet man üblicherweise das Dateisystem NTFS.

Heutige Gründe für Partitionierungen:

- Installation mehrerer Betriebssysteme
- Einrichten verschiedener Dateisysteme
- Reservierung für eine Windows- oder Linux-Auslagerungsdatei (SWAP-Partition)
- Installationsdateien
- sehr großen Speicher verwalten
- Trennung von Programmen und Daten

5.6.1.3) Logische Formatierung (High-Level-Formatierung)

Ein physikalisch formatiertes Festplattenlaufwerk muss noch logisch formatiert werden. Durch die logische Formatierung wird ein **Dateisystem** auf der Festplatte definiert. Erst ein Dateisystem gestattet es einem Betriebssystem wie z.B. DOS, OS/2, Windows 95/98, Windows NT/Windows 2000/WindowsXP/Windows7/Windows8 oder Linux), den verfügbaren Speicher für das Speichern und Laden von Dateien zu nutzen.

Vor der logischen Formatierung kann eine Festplatte in **Partitionen** unterteilt werden. Auf jeder Partition kann ein Dateisystem (logisches Format) definiert werden. Eine Festplattenpartition, die bereits logisch formatiert ist, wird als Datenträger (Volume) bezeichnet. Als Teil des Formatierungsvorgangs werden Sie durch das Formatierungsprogramm aufgefordert, der Partition einen Namen zuzuweisen, die sogenannte "Datenträgerbezeichnung". Mit diesem Namen können Sie den Datenträger (die Partition) später identifizieren.

Normal- und Schnellformatierung

Es gibt zwei unterschiedliche Methoden, den Datenträger high-level zu formatieren:

Die Schnell- und die Normalformatierung.

Sie unterscheiden sich in folgenden Punkten:

Normalformatierung – Es wird eine Suche nach fehlerhaften Sektoren durchgeführt. Diese Suche nimmt den Großteil der Zeit in Anspruch. Anschließend erfolgt das Schreiben der Dateisystem-Metadaten und somit die Löschung der vorhandenen Dateien.

Schnellformatierung – Es werden zwar die Dateien aus dem Inhaltsverzeichnis des Datenträgers beziehungsweise der betreffenden Partition entfernt, die Suche nach fehlerhaften Sektoren wird jedoch ausgelassen.

Bei einer High-Level-Formatierung sind in dem neuen Dateisystem die alten Daten nicht verfügbar, da sie nicht mehr durch entsprechende Verweise im Dateisystem referenziert werden. Sie werden jedoch nicht notwendigerweise gelöscht. Meistens verbleiben sie rein physikalisch auf der Festplatte, bis sie mit neuen Daten überschrieben werden. Solange die verwendeten Datenblöcke nicht erneut beschrieben werden, ist mit entsprechender Software eine weitgehende Wiederherstellung noch möglich, gestaltet sich jedoch schwerer, als wenn die Dateien einfach nur gelöscht würden. Nicht selten können daher via Software nur die einzelnen Dateien, nicht aber die Ordnerstruktur wiederhergestellt werden.

Dateisysteme

Jeder Computer benötigt ein funktionsfähiges Betriebssystem, auch wenn es lediglich aus den DOS- oder Windows 98-Bootdateien besteht. Verschiedene Betriebssysteme verwenden unterschiedliche Dateisysteme, um auf Datenträger zuzugreifen und die Speicherorte von Daten zu ermitteln. Das Dateisystem bestimmt hierbei die Art der Verwaltung und des Zugriffs auf Dateien auf einem Datenträger.

Die Methoden und Datenstrukturen, die ein Betriebssystem verwendet, um Speicherorte von Dateien auf einer Partition zu ermitteln, werden also als das **Dateisystem** definiert. Bei den hier besprochenen Dateisystemen handelt es sich um FAT, FAT32, HPFS, NTFS, ReFS, Linux Ext2 und Linux ReiserFS.

Windows

FAT16

Das FAT16-Dateisystem ist das ursprünglich von MS-DOS für die Dateiverwaltung verwendete System. Das FAT (File Allocation Table) ist eine Datenstruktur, die MS-DOS beim Formatieren eines Datenträgers auf demselben erstellt. FAT16-Partitionen können bis zu 2 GB groß sein. Die Größe einer Partition bestimmt die Größe der Cluster, in denen Daten gespeichert werden. Bei einer Partitionsgröße von über einem GB kann dies zu einem erheblichen Speicherverlust führen. FAT16-Partitionen werden bei den meisten Versionen von DOS und Windows 95 eingesetzt und von den folgenden Betriebssystemen unterstützt: Windows 95, Windows 98, Windows NT, Windows 2000, DOS (inklusive MS-DOS, PC-DOS und DR-DOS), OS/2, Linux und viele andere Betriebssysteme.

FAT32

FAT32 ist ein Dateisystem, das von Microsoft für Windows 95, Version B (OSR2), entwickelt wurde. Es verfügt über die Fähigkeit, 32-Bit-Clusteradressen zu verwalten und unterstützt Partitionen von bis zu 8 GB bei einer Clustergröße von nur 4 KB. Partitionen von bis zu 16 GB können bei einer Clustergröße von 8 KB verwaltet werden, was den unnötigen Verbrauch von Festplattenspeicher aufgrund ineffizienter Clusternutzung deutlich reduzieren kann. FAT32-Partitionen kommen ausschließlich unter Windows 95B oder höher, Windows 98, Windows 2000 und unter neueren Versionen von Linux zum Einsatz.

NTFS

NTFS (New Technology File System) wurde von Microsoft speziell für die Verwendung mit Windows NT, Windows 2000 und Windows XP entwickelt und unterstützt die verbesserten Sicherheitsmerkmale, die über dieses Betriebssystem zur Verfügung stehen. NTFS unterstützt vollständige Zugriffskontrolle, Dateisystemwiederherstellung und besonders große Datenträger. NTFS-Partitionen werden ausschließlich unter Windows NT (Workstation und Server) sowie unter Windows 2000, Windows XP, Windows 7 und Windows 8 eingesetzt.

ReFS

ReFS (Resilient File System) ist ein Dateisystem von Microsoft, wobei engl. „resilient“ für „elastisch“, „belastbar“ oder gar „unverwüstlich“ steht. Es wurde mit den Betriebssystemen Windows 8 und Windows Server 2012 als Ergänzung zum NTFS-Dateisystem eingeführt.

Linux

Ext2

Das Ext2-Dateisystem war bis ca. 2002 das Standardsystem für Linux. Ist außerordentlich stabil und sicher, aber leider muss nach einem Systemabsturz (z.B. Stromausfall) eine sehr zeitaufwändige Überprüfung des gesamten Dateisystems durchgeführt werden.

Ext3

Das Ext3-Dateisystem ist z.Z. das Standardsystem für Linux. Weitgehend kompatibel zu Ext2, hat aber den Vorteil, dass es über eine **Journaling-Funktion** verfügt. D.h. nach einem Stromausfall entfällt eine langwierige Überprüfung des gesamten Dateisystems.

Ext4

Ext4 (Fourth Extended File System): weiterentwickelte Variante von Ext3, unter anderem mit

erweiterten Limits.

ReiserFS

Das ReiserFS ist eine Weiterentwicklung des Ext2-Dateisystems ist das Standardsystem für Linux (Verbesserte Dateisystemwiederherstellung nach Unterbrechungen im laufenden Betrieb). Linux ReiserFS-Partitionen werden ausschließlich unter Linux-Installationen verwendet.

MAC

HFS

- [W HFS](#)

Weiterführende Informationen

Festplatten Fakten

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_06

Last update: **2019/05/14 15:49**



5.7) Planung eines Systems

Man sollte sich vor der Installation der Betriebssysteme (oder eines Bootmanagers) einige Gedanken über den Aufbau des Systems machen: Hierbei ist am wichtigsten, sich zuerst überlegen, welche Betriebssysteme eingesetzt werden sollen.

Als nächstes ist abzuklären, wie viel Festplattenplatz für welches Betriebssystem benötigt wird (Größe der Festplatte). Hierzu macht der Hersteller des Betriebssystems bereits einen Vorschlag, der als Minimum eingehalten werden sollte.

Unter Umständen kann es sinnvoll sein, einen Teil der Festplatte (eine Partition) ausschließlich dazu zu verwenden, Daten unter allen, oder zumindest unter mehreren Betriebssystemen, zur Verfügung zu stellen. Am besten lässt man einen Teil der Festplatte unbenutzt (falls diese groß genug ist), damit dieser später für Erweiterungen genutzt werden kann.

Besonderheiten und Probleme der einzelnen Betriebssysteme

Probleme mit DOS / Windows 95/98/ME

Bei der Verwendung von FAT 16 durfte die Partition nicht größer als 2 GB sein.

Von der 2. Festplatte kann nur gebootet werden, wenn auf der 1. Festplatte keine primäre Partition sichtbar ist.

Probleme mit Windows NT/2000/XP

- Windows 2000/XP: Das Setup-Programm von Windows 2000/XP überschreibt während der Installation den Master-Boot-Record.
- Unter Windows 2000/XP kann eine einmal angelegte Partition nicht mehr verkleinert werden.

Windows 7, Windows 8

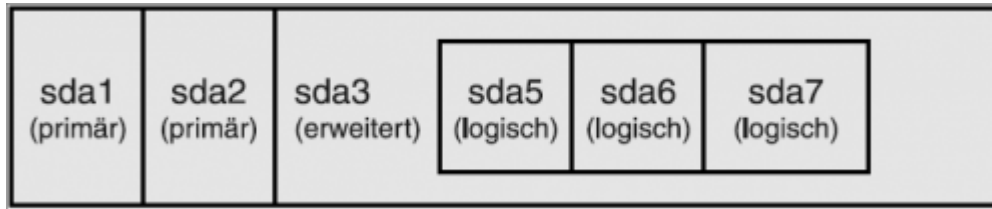
- ab Windows 7 können Partitionen auch verkleinert werden:
 - Datenträgerverwaltung - Rechtsklick auf die Partition - Volume verkleinern

Linux

Linux kann auf jeder Festplatte maximal 15 Partitionen ansprechen, davon maximal 11 logische Partitionen.

Unter Linux erfolgt der interne Zugriff auf Festplatten bzw. deren Partitionen über so genannte Device-Dateien: Die Festplatten erhalten der Reihe nach die Bezeichnungen /dev/sda, /dev/sdb, /dev/sdc etc.

Um eine einzelne Partition und nicht die ganze Festplatte anzusprechen, wird der Name um die Partitionsnummer ergänzt. Die Zahlen 1 bis 4 sind für primäre und erweiterte Partitionen reserviert. Logische Partitionen beginnen mit der Nummer 5 – auch dann, wenn es weniger als vier primäre oder erweiterte Partitionen gibt. Die folgende Abbildung veranschaulicht die Nummerierung: Auf der Festplatte gibt es zwei primäre Partitionen und eine erweiterte Partition, die drei logische Partitionen enthält.



Die maximale Partitionsgröße beträgt 2 TByte. Da es mittlerweile Festplatten mit mehr als 2 TByte Speichervolumen gibt, ist eine sinnvolle Nutzung von Festplatten mit mehr als 2 TByte nur noch mit GPT-Partitionstabellen möglich.

Anzahl und Größe der Linux-Partitionen

Leider gibt es auf die Frage, wie viele Linux-Partitionen ein System haben sollte, keine allgemein gültige Antwort.

Die **Systempartition** ist die einzige Partition, die man unbedingt benötigt. Sie nimmt das Linux-System mit all seinen Programmen auf. Diese Partition bekommt immer den Namen /. Dabei handelt es sich genau genommen um den Punkt, an dem die Partition in das Dateisystem eingebunden wird (den mount-Punkt).

Eine vernünftige Größe für die Installation und den Betrieb einer gängigen Distribution liegt bei 10 bis 20 GByte.

Mit einer **Datenpartition** trennt man den Speicherort für die Systemdateien und für die eigenen Dateien. Das hat einen wesentlichen Vorteil: Man kann später problemlos eine neue Distribution in die Systempartition installieren, ohne die davon getrennte Datenpartition mit Ihren eigenen Daten zu gefährden. Bei der Datenpartition wird **/home** als Name bzw. mount-Punkt verwendet, weswegen oft auch von einer Home-Partition die Rede ist. Die Größe hängt von den eigenen Bedürfnissen ab.

Die **Swap-Partition** ist das Gegenstück zur Auslagerungsdatei von Windows: Wenn Linux zu wenig RAM hat, lagert es Teile des gerade nicht benötigten RAM-Inhalts dorthin aus. Im Gegensatz zu den anderen Partitionen bekommt die Swap-Partition keinen Namen (keinen mount-Punkt). Die Größe sollte das ein- bis zweifache des RAMs betragen.

Mehrere Betriebssysteme verwalten

Es gibt verschiedene Möglichkeiten, mehrere Betriebssysteme zu verwalten:

- Mit Hilfe von Bootmanagementprogrammen, wie beispielsweise BootMagic von PowerQuest, Bootstar von Star-Tools, Acronis von SWSOft oder GRUB von SuSE-Linux
- Mit Hilfe einer von einem Betriebssystem verwalteten doppelten Bootkonfiguration, wie

beispielsweise der Boot Loader von Windows NT oder GRUB (bzw. mit dem älteren) LILO unter Linux.

- Manuell, indem Sie ein Betriebssystem als „aktiv“ einstellen. Nehmen Sie diese Einstellung über ein Dienstprogramm wie PQBoot vor, oder bearbeiten Sie den Masterbootdatensatz von Hand (nicht zu empfehlen!).

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_07



Last update: **2019/05/14 15:49**

5.8) Verschiedene Betriebssystem

Windows-Betriebssysteme

- [W MS-DOS / DOS](#)
- Windows 3.11
- Windows 95
- Windows 98
- Windows NT
- Windows 2000
- [W Windows XP](#)
- [Windows Vista](#)
- [W Windows 7](#)
- [W Windows 8](#)
- [W Windows 10](#)

Linux-Betriebssysteme

- [OpenSuSE Linux](#)
 - [Open Suse für Profis](#)
- [SuSE Linux](#)

Ubuntu

- [Ubuntu](#)
- [www.ubuntu.com](#)
 - [Ubuntu Austria](#)
 - [Ubuntu Server](#)
- Kubuntu
- Edubuntu
- Xubuntu
- Gobuntu
- [Knoppix](#)
- RedHat Linux
- Debian Linux
- Xubuntu von USB-Stick: [Pendrive Linux](#)

Macintosh-Betriebssysteme

- [Leopard](#)
- [Snow Leopard](#)

Betriebssystem-Virtualisierung

- [Virtualisierung](#)

Online-Betriebssysteme

- [Ein Online-Desktop](#) von [eyeOS](#)

Portable-Betriebssysteme

- [PC am Datenstick](#)
- [Linux Advanced](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_08



Last update: **2019/05/14 15:50**

5.9) Windows

Microsoft Windows bzw. Windows ist **ursprünglich** eine von Microsoft entwickelte **grafische Benutzeroberfläche**, aus der **später** eine Reihe von eigenständigen **Betriebssystemen** entstanden sind.

Das ursprüngliche Windows war eine **grafische Erweiterung des Betriebssystems MS-DOS**, wie beispielsweise auch GEM oder PC/GEOS. Dieses System wurde ab Windows 95 um einen überarbeiteten Kernel, die 32-Bit-API Win32 und Internetfähigkeit erweitert und mit den Versionen Windows 98, 98 SE sowie ME weitergeführt. Für diese Betriebssystemfamilie hat sich die Sammelbezeichnung Windows 9x etabliert. Parallel dazu wurde unter der Leitung von David N. Cutler seit 1988 das auf den Konzepten des Betriebssystems VMS basierende Windows NT entwickelt. Seit Windows XP vertriebt Microsoft für den Desktop nur noch Nachfolger von Windows NT, da Windows 9x aufgrund technischer Schwächen aufgegeben wurde. Seitdem heißt das Betriebssystem als Ganzes nur noch Windows (englisch für „Fenster“). Der Name rührt daher, dass die Benutzeroberflächen von Anwendungen auf dem Bildschirm als rechteckige Fenster dargestellt werden.

Windows-Betriebssysteme sind vor allem auf **Personal Computern und Servern verbreitet**; daneben existieren Varianten für Geräte wie Smartphones oder PDAs sowie für spezielle Embedded Devices wie etwa vollelektronische Messgeräte und Einzelhandels-Kassensysteme oder für die Anwendung in Kraftfahrzeugen. Kevin Turner, der Chief Operating Officer von Microsoft, nannte auf der Worldwide Partner Conference 2014 einen Gesamt-Marktanteil von 14 Prozent für alle Windows-Varianten.

- [Kommandozeile](#)
- [5.9.1\) MS-DOS](#)
- [5.9.2\) Benutzerverwaltung](#)
- [5.9.3\) Dateimanagement](#)
- [5.9.4\) Sicherung & Wiederherstellung](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_09



Last update: **2019/05/14 15:53**

5.9.1) Das Betriebssystem MSDOS

MSDOS (DOS = Disk Operating System) ist ein Betriebssystem, das von der Firma Microsoft in den USA entwickelt wurde und früher zu den am meisten verbreiteten Betriebssystemen zählte. Die Entwicklungsgeschichte reicht sehr lange zurück, die erste Version gab es 1981.

Mittlerweile ist MSDOS weitgehend durch die neuen Windows-Betriebssysteme abgelöst worden - trotzdem wird DOS für bestimmte Aufgaben auch heute noch verwendet, wie z.B. zur Erstellung von Batch-Dateien. Die Kenntnisse über DOS könnten auch insbesondere bei einem Crash nützlich sein, wenn Windows z.B. nicht mehr startet oder man ein Uralt-DOS-PC-Spiel noch mal spielen will.

Arbeiten mit DOS auf der Systemkonsole

Speicher zuvor folgenden Ordner auf deinem Verzeichnis: [dos-befehle.zip](#)

Start der Systemkonsole

- Start - Ausführen: cmd
- Start - Programme - Zubehör - Eingabeaufforderung
- Vollbildmodus: ALT + ENTER
- Beenden: exit

Einige wichtige DOS-Befehle

Allgemeine Befehle

Befehl	Beschreibung
help	zeigt eine Liste aller MS-DOS-Befehle und eine kurze Beschreibung
help <i>Befehl</i>	man erhält Informationen zu einem bestimmten Befehl
dir	zeigt Inhaltsverzeichnis des aktuellen Verzeichnisses an
dir i:	zeigt den Inhalt des Laufwerks i
dir *.exe	Stern ist eine „Wild Card“. Er steht für eine beliebige Zeichenkombination, es werden somit alle .EXE-Dateien ausgegeben
dir m??er.doc	Fragezeichen ist ebenfalls eine „Wild Card“, jedoch für ein einzelnes Zeichen
whoami	Gibt den Usernamen des ausführenden Benutzers aus
hostname	Gibt den Rechnernamen aus

Befehle für Verzeichnisse

Die folgende Befehle sind zum Anlegen, Wechseln und Löschen neuer Verzeichnisse. Das Verzeichnis, welches im aktuellen Eingabebereitschaftszeichen aufscheint, wird meist als **aktuelles Verzeichnis** oder **Arbeitsverzeichnis** bezeichnet.

Befehl	Beschreibung
cd	change directory, Bedeutung von Syntax abhängig
cd \.	geht zum root (oberste Verzeichnisebene)
cd \programme	wechselt in das Verzeichnis Programme ausgehend vom root
cd programme	wechselt - ausgehend vom derzeitigen Arbeitsverzeichnis - in das Unterverzeichnis programme
cd .	Zeiger auf aktuelles Verzeichnis
cd ..	Zeiger auf übergeordnetes Verzeichnis
md	make directory - erzeugt Unterverzeichnis
rd	remove directory - löscht Unterverzeichnis
tree	zeigt die Verzeichnisstruktur auf dem aktuellen Laufwerk an

Befehle für Dateien

Wichtiger Operationen mit Dateien sind das Kopieren, das Löschen und das Umbenennen.

Befehl	Beschreibung
copy	benötigt zwei Parameter: 1. Quelldatei, die kopiert werden soll, 2. Zieldatei
copy x.dat x.old	kopiert die Datei x.dat auf x.old (dabei wird x.dat nicht gelöscht!). Existiert bereits eine Datei x.old, so wird diese gelöscht und mit den Daten von x.dat gefüllt.
copy *.* a:	kopiert alle Dateien im Arbeitsverzeichnis nach A:
copy adam.dat a:	kopiert adam.dat nach a:adam.dat
copy *.* c:\xy\z	kopiert alle Dateien des aktuellen Ordners auf die Festplatte c: ins Unterverzeichnis Z des Verzeichnisses xy
del x.dat	löscht die Datei x . dat im aktuellen Verzeichnis
del *.*	löscht alle Dateien im aktuellen Verzeichnis
ren x.dat y.dat	rename, benennt x.dat in y.dat um
move muster.doc windows	verschiebt muster . doc ind das Verzeichnis windows
type Dateiname	zeigt den Inhalt der angegebenen Datei am Bildschirm an
edit Dateiname	ruft einen Editor auf, um den Inhalt einer Datei verändern zu können
print Dateiname	gibt den Inhalt einer Datei auf dem Drucker aus
xcopy /E	Kopiert ganze Verzeichnisse
rmdir /S	Löscht Verzeichnisse trotz Inhalt

DOS-Übung

Speichere den Ordner „DOS“ [dos.rar](#) in dein Verzeichnis, bevor du mit der Übung beginnst!

1. Erzeuge ein Verzeichnis uebungen in deinem homedirectory.
2. Wechsle in dieses Verzeichnis.
3. Erstelle eine Datei test1.txt und test2.txt mit beliebigen Inhalt.
4. Kopiere die Datei test1.txt in die Datei test21.txt.
5. Erstelle ein Verzeichnis aufgabe.
6. Benenne test2.txt um in test12.txt.
7. Kopiere die Dateien test1?.txt in das Verzeichnis aufgabe. Welche Dateien wurden kopiert?
8. Kopiere dir alle Dateien aus dem OrdnerDOS in das Verzeichnis aufgabe. Wie viele Dateien

befinden sich nun in diesem Verzeichnis?

9. Lasse dir alle Dateien mit der Endung *.txt ausgeben.
10. Lösche alle Dateien mit der Endung *.dat.
11. Benenne alle Dateien mit der Endung *.bak um in *.txt.
12. Lasse dir alle Dateien sortiert nach der Dateigröße anzeigen. Finde den Befehl selber heraus.
13. Gib den Inhalt der Datei hallo.txt aus.
14. Lasse dir alle Dateien anzeigen, bei denen im Dateinamen die Zahl 1 vorkommt.
15. Lösche diese Dateien.
16. Verschiebe alle Dateien aus dem Verzeichnis Aufgabe in das Verzeichnis uebungen.
17. Lösche das Verzeichnis Aufgabe.
18. Gib den Befehl `attrib +h *.*` ein.
19. Lasse dir alle Dateien anzeigen. Was ist passiert?
20. Mache dir über den Befehl `attrib` schlau und gib einen Befehl ein, sodass alle Dateien wieder angezeigt werden.

BATCH (.bat) Files

Befehl	Beschreibung
echo	Gibt Text aus
@echo off	Schaltet die Autobefehlsanzeige aus
title	Definiert den Titel des BATCH-Files
set VAR=Text	Speichert Text in die Variable VAR
set /P VAR=	Fordert den User auf eine Zeichenfolge einzugeben
set/A C=%A%+%B%	/A gibt an, dass die Zeichenfolge rechts vom = ein numerischer Ausdruck ist, der ausgewertet wird. in C wird das Ergebnis von A+B gespeichert
echo %VAR%	Ausgabe einer Variable → Text
echo %USERNAME%	Gibt den Usernamen des ausführenden Benutzers aus
REM	Definiert einen Kommentar

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_09:05_09_01



Last update: **2019/05/14 15:53**

5.9.2) Benutzerverwaltung

Microsoft Windows ist ein Multi-User Betriebssystem. Was ist ein Benutzer? Was ist eine Gruppe? Welche Rolle spielt die Sicherheitsrichtlinie?

BENUTZER

SID

Jeder Benutzer wird durch eine Nummer, einer SID (Security Identifier) eindeutig gekennzeichnet und erhält ein eigenes Profil unter `c:\users\%username%`.

Mithilfe von SIDs ist es u.a. möglich Benutzer umzubenennen, ohne Veränderung seiner Zugriffsrechte. Selbst wenn das Administratorkonto umbenannt wird, so hat dieser immer die SID mit der Endung 500. Welcher User welche SID verwendet kann in der CMD oder mit PowerShell eingesehen werden:

Befehl für die PowerShell:

```
PS C:\> get-wmiobject win32_useraccount

AccountType : 512
Caption     : SERVER01\Administrator
Domain      : SERVER01
SID         : S-1-5-21-140281148-68646805-4244902722-500
FullName    :
Name        : Administrator

AccountType : 512
Caption     : SERVER01\Gast
Domain      : SERVER01
SID         : S-1-5-21-140281148-68646805-4244902722-501
FullName    :
Name        : Gast
```

Befehl für die CMD:

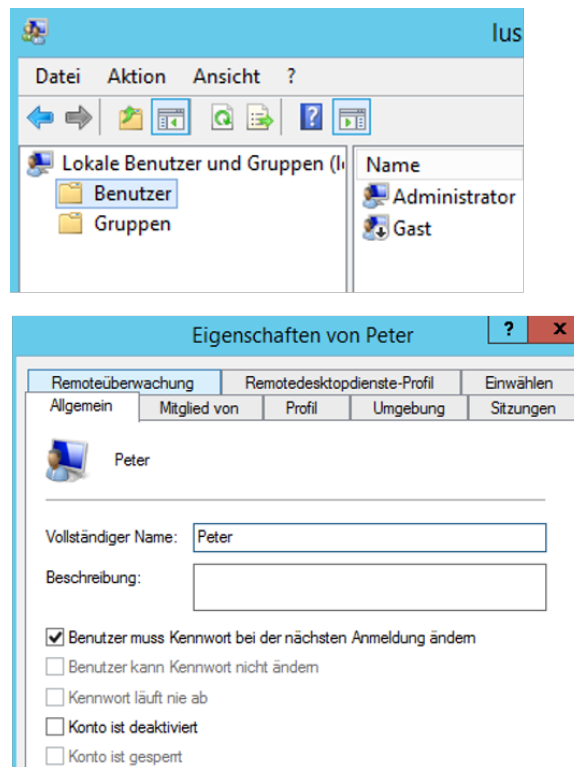
```
wmic useraccount get name,sid
```

Kennwörter/Passwörter

Kennwörter werden in der SAM (Security Account Manager) Datenbank gespeichert. Diese Datenbank kann im laufenden Betrieb (mit Bordmitteln) nicht eingesehen werden. Die Kennwörter werden als Hashwert gespeichert, d.h. nicht im Klartext.

Anlegen von Lokalen Benutzern

Dies ist grafisch mit lusrmgr.msc oder mithilfe des Befehls net user möglich.



```
net user Max 'Pa$$w0rd' /add
```

Massenimport von Usern

Hier am Beispiel einer Textdatei users.txt mit einer Auflistung von Vornamen:

```

C:\>FOR /F %i in (users.txt) DO NET USER %i Pa$$w0rd /add

C:\>NET USER leo Pa$$w0rd /add
Der Befehl wurde erfolgreich ausgeführt.

C:\>NET USER sandra Pa$$w0rd /add
Der Befehl wurde erfolgreich ausgeführt.

C:\>NET USER patrick Pa$$w0rd /add
Der Befehl wurde erfolgreich ausgeführt.

C:\>NET USER karl Pa$$w0rd /add
Der Befehl wurde erfolgreich ausgeführt.

PS C:\> get-content .\users.txt | ForEach-Object -Process {net user $_ /delete}
Der Befehl wurde erfolgreich ausgeführt.
Der Befehl wurde erfolgreich ausgeführt.
Der Befehl wurde erfolgreich ausgeführt.
oder /add

```

GRUPPEN

Bei einer Gruppe handelt es sich um eine Sammlung von Benutzer- und Computerkonten, Kontakten sowie weiteren Gruppen, die als einzelne Einheit verwaltet werden können. Gruppen findet man ebenfalls in lusrmgr.msc bzw. auch mit net localgroup.

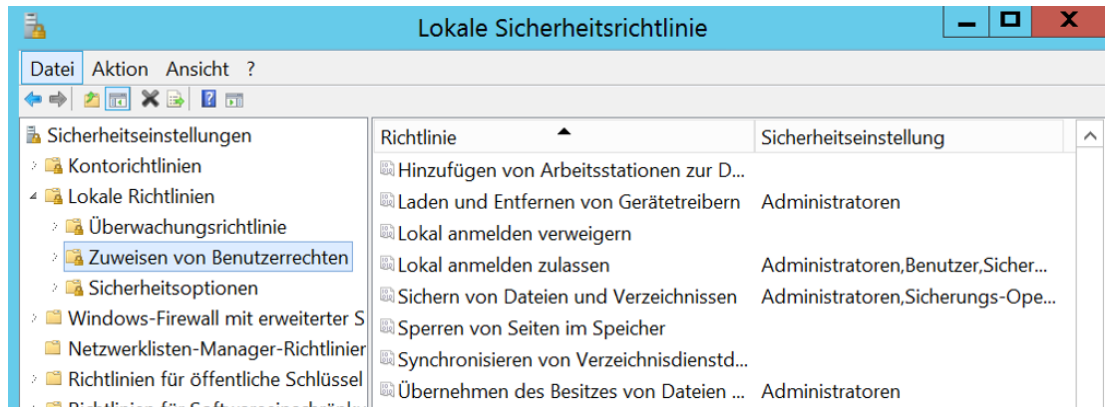
Lokale Benutzer und Gruppen	
Name	Beschreibung
Administratoren	Administratoren haben uneingeschränkten Vollzugriff auf den Computer bzw. die
Benutzer	Benutzer können keine zufälligen oder beabsichtigten Änderungen am System durchführen
Distributed COM-Benutzer	Mitglieder dieser Gruppe können Distributed-COM-Objekte auf diesem Computer verwenden
Druck-Operatoren	Mitglieder dieser Gruppe können Drucker in der Domäne verwalten.
Ereignisprotokollleser	Mitglieder dieser Gruppe dürfen Ereignisprotokolle des lokalen Computers lesen

Hier wird der Benutzer admin01 der Gruppe Administratoren hinzugefügt:

```
net localgroup administratoren admin01 /add
```

SICHERHEITSRICHTLINIEN

Wozu das Ganze? Gruppen, Benutzer? Nun, es ist wichtig zu unterscheiden, wer, was, wann, wo am System tun darf oder nicht. Und dies regelt die Sicherheitsrichtlinie, welche unter secpol.msc zu finden ist.



Fügen Sie einen Benutzer in die Gruppe der Administratoren hinzu, so darf dieser laut der Liste Gerätetreiber installieren oder auch Dateien sichern. Würde es keine Gruppen(-richtlinien) geben, so müsste man bei jedem einzelnen Benutzer jedes einzelne Recht hinzufügen oder entfernen. Mithilfe von Gruppen braucht man dies nur einmal für die Gruppe und kann später alle gewünschten Benutzer in diese Gruppe hinzufügen. Somit spart man sehr viel Zeit als IT-Administrator.

From:
<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_09:05_09_02



Last update: **2019/05/14 15:54**

5.9.3) Dateimanagement

Eine Hauptaufgabe des Betriebssystems ist die Verwaltung der Daten. Dabei werden Daten in sogenannten **Dateien** zusammengefasst, die auf Festplatten und anderen Speichermedien abgespeichert werden können. Eine Datei lässt sich mit dem Inhalt einer Karteikarte vergleichen.

Mehrere Dateien („Karteikarten“) werden in einem Verzeichnis (directory) abgelegt. Ein Verzeichnis ist vergleichbar mit einer Schachtel, in der verschiedene Karteikarten untergebracht werden können. Selbstverständlich ist es auch möglich, in einer Schachtel mehrere kleinere Schachteln unterzubringen (Unterverzeichnisse - subdirectory) usw.

Klarerweise entsteht dadurch ein baumartiges System, weshalb man dieses auch als „Tree“ bezeichnet. Die größte Schachtel, die alle anderen enthält bezeichnet man als **Hauptverzeichnis** oder **Root** (Wurzel).

Dateiformate

Die Art und Weise, wie Informationen innerhalb einer Datei gespeichert werden, und wie diese zu interpretieren sind, wird als Dateiformat bezeichnet. Um den Inhalt einer Datei wieder benutzen oder weiterverwenden zu können, muss bekannt sein, auf welche Weise und in welcher Reihenfolge die Informationen in der Datei abgelegt wurden.

Für unterschiedliche Inhalte und Einsatzzwecke gibt es zahlreiche Methoden zur Speicherung in Dateien, die unter dem Oberbegriff **Dateiformate** zusammengefasst werden. Um also z.B. den Inhalt eines einfachen Briefes in einer Datei abzulegen, wird ein auf Textspeicherung ausgerichtetes Format verwendet, während bei der Speicherung eines Bildes ein entsprechendes Grafikformat verwendet wird.

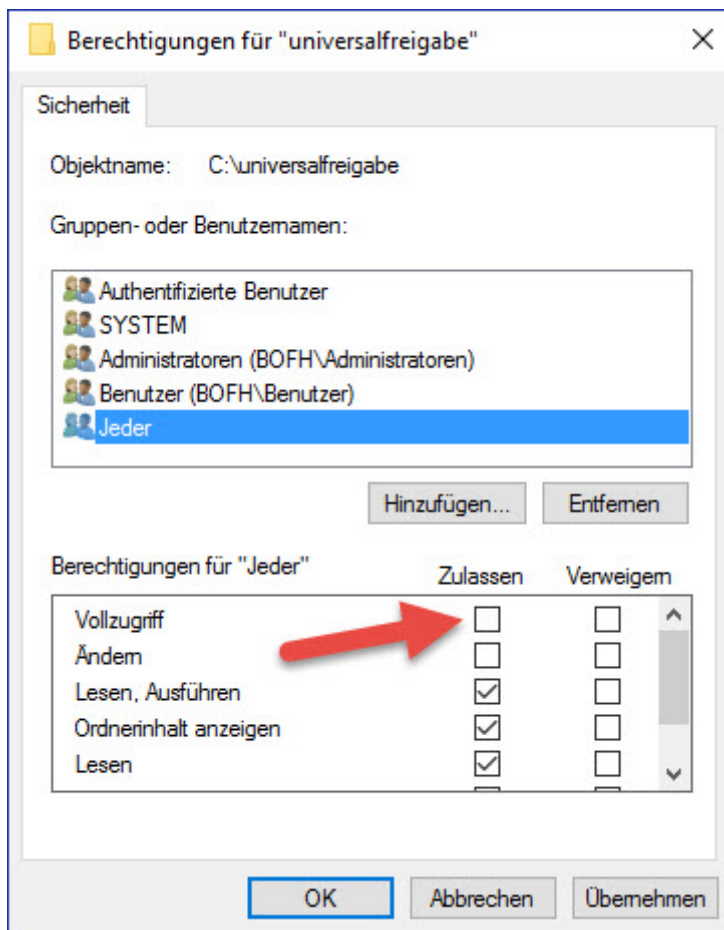
Vor allem bei Microsoft-Betriebssystemen hat es sich eingebürgert, einen Hinweis auf das verwendete Dateiformat durch einen Zusatz am jeweiligen Namen der Datei anzubringen. Diesen Hinweis nennt man **Dateiendung** und wird durch einen **Punkt** vom Dateinamen getrennt.

Übersicht über einige Standarderweiterungen

für ausführbare Dateien	
EXE	ausführbare Datei
BAT	Batch-File
programmspezifische Erweiterungen	
DOC	formatiertes Dokument
TXT	Textdatei ohne Sonderzeichen
SYS	System(Programm)datei
DBF	Datenbankfile
...	...

Dateirechte

Da nicht jeder Benutzer immer alle Rechte auf einem Rechner haben darf/soll/muss, gibt es im Dateisystem (z.B. NTFS) die Rechte für die jeweiligen Benutzer bzw. Gruppen zu definieren. Diese Dateirechte sind die Sicherheitseinstellungen, die man auf einer NTFS-Partition einem Dateiojekt vergibt. Diese werden üblicherweise im Kontextmenü des Objektes unter Eigenschaften > Sicherheitseinstellungen vergeben:



Es gibt die folgenden Rechte in der Registerkarte Sicherheit bei einem Dateiojekt

Berechtigung	Rechte
Vollzugriff / Full Control	Alle
Ändern / Modify	Alle, außer Berechtigungen ändern und Besitzerrechte übernehmen
Lesen & Ausführen / Read & Execute	Datei öffnen und lesen, ausführbare Dateien und Batchdateien starten
Ordnerinhalt auflisten / List Folder Contents	Nur bei Ordner: Lesen und Lesen & Ausführen
Lesen / Read	Datei öffnen und lesen
Schreiben / Write	Datei ändern oder neu erzeugen
Spezielle Berechtigungen	Siehe unten

Kopieren / Verschieben von Dateien

Wenn eine Datei kopiert wird, wird sie am Zielort neu erstellt und erbt daher die Berechtigungen des Ordners, in den sie kopiert wurde. Beim Verschieben innerhalb einer Partition behält die Datei Ihre Ursprungsrechte.

Will man Dateien verschieben und Berechtigungen des Zielordners annehmen lassen, kopiert man also am Besten und löscht dann die Ursprungsdaten.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_09:05_09_03



Last update: **2019/05/14 15:55**

5.9.4) Sichern und Wiederherstellen in Windows

Ein Windows 10 Backup könnt ihr mit integrierten Backup-Tools im Handumdrehen erstellen, eure persönlichen Daten durch eine Sicherung schützen und ein komplettes Windows-Systemabbild mit Bordmitteln erstellen.

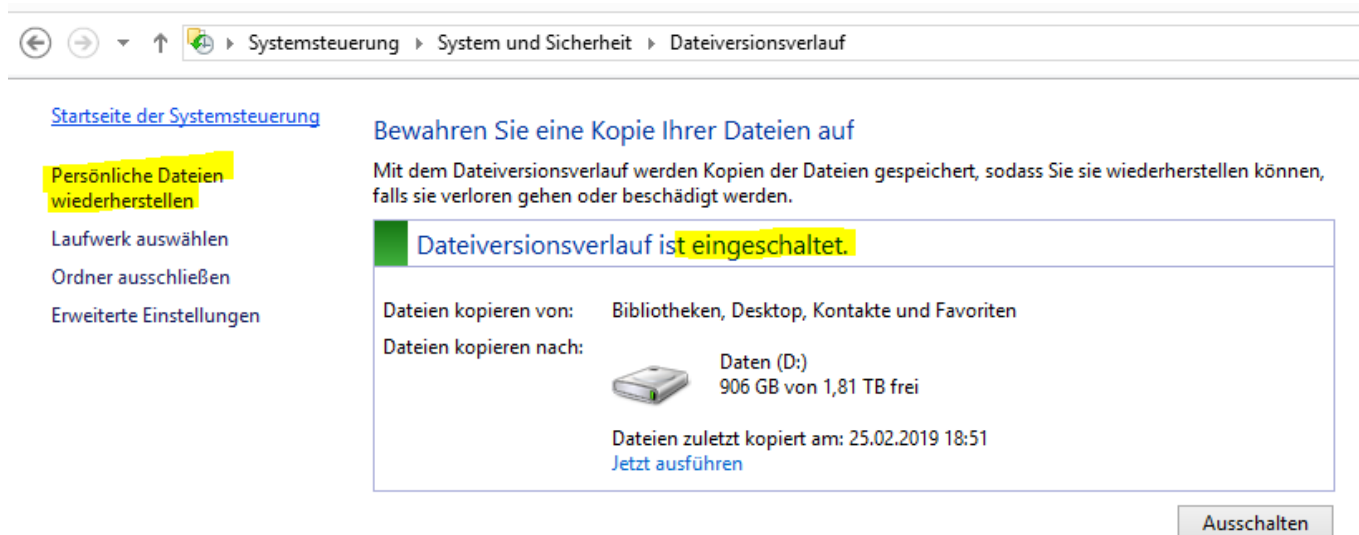
Mit den Windows 10-Bordmitteln könnt ihr bei einem Defekt der Festplatte nicht nur einzelne Daten, sondern euer komplettes Windows-System wiederherstellen. Außerdem könnt ihr mit den Windows-Systemprogrammen eure Daten automatisch sichern.

Hier sind die einzelnen Varianten beschrieben:

1) Dateiversionsverlauf

Mit dem Dateiversionsverlauf werden Kopien der Dateien gespeichert, sodass sie wiederhergestellt werden können, falls sie verloren gehen oder beschädigt werden.

- Systemsteuerung - System und Sicherheit - Dateiversionsverlauf
- Wenn der Dateiversionsverlauf eingeschaltet ist, kann auf ältere Dateibestände zurückgegriffen werden.



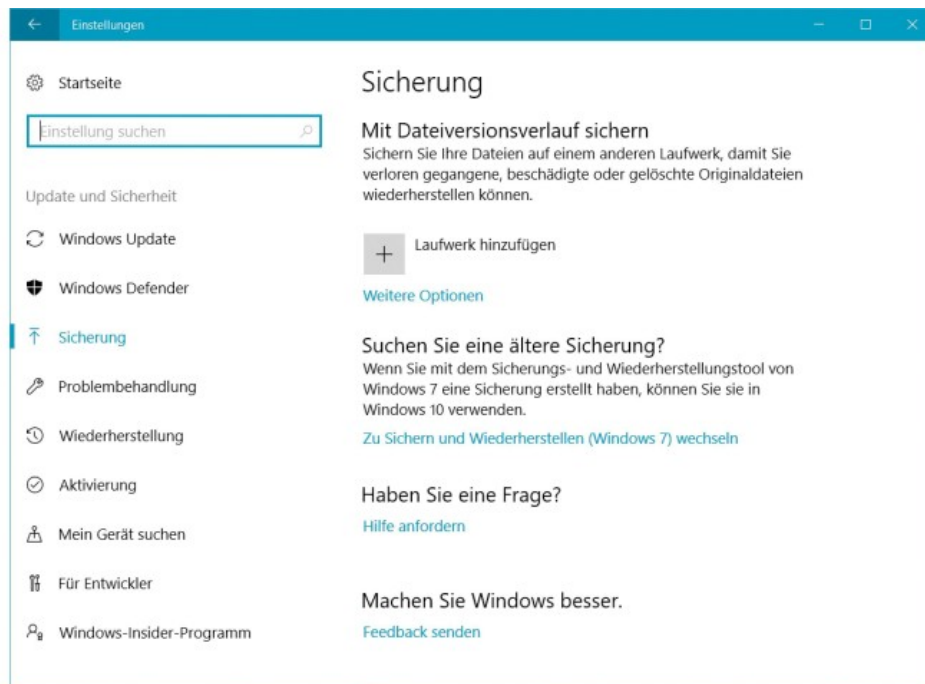
2) Backup erstellen und einzelne Dateien sichern

Mit den folgenden Schritten könnt ihr einzelne Ordner und Dateien, etwa Bilder, Videos und Dokumente, automatisch sichern und einzelne Dateien im Fehlerfall wiederherstellen:

- Öffnet unten links das vierkachelige Windows-Startmenü und klickt auf den Reiter „Einstellungen“.
- Wählt im neuen Fenster das Feld „Update und Sicherheit“ und klickt anschließend auf die

Kategorie „Sicherung“.

- Klickt auf „Laufwerk hinzufügen“ und Windows listet euch alle verfügbaren Datenträger aus. Wählt die Festplatte oder den USB-Stick aus, auf dem eure Daten gesichert werden sollen.
- Klickt danach auf „Weitere Optionen“, wo ihr über „Ordner hinzufügen“ weitere Ordner und Daten dem Backup hinzufügen könnt. Über das Drop-Down-Menü bei „Meine Daten sichern“ könnt ihr außerdem auswählen, ob und wie oft eure Daten automatisch gesichert werden sollen.
- Schießt die Datensicherung mit einem Klick auf „Jetzt sichern“ ab.

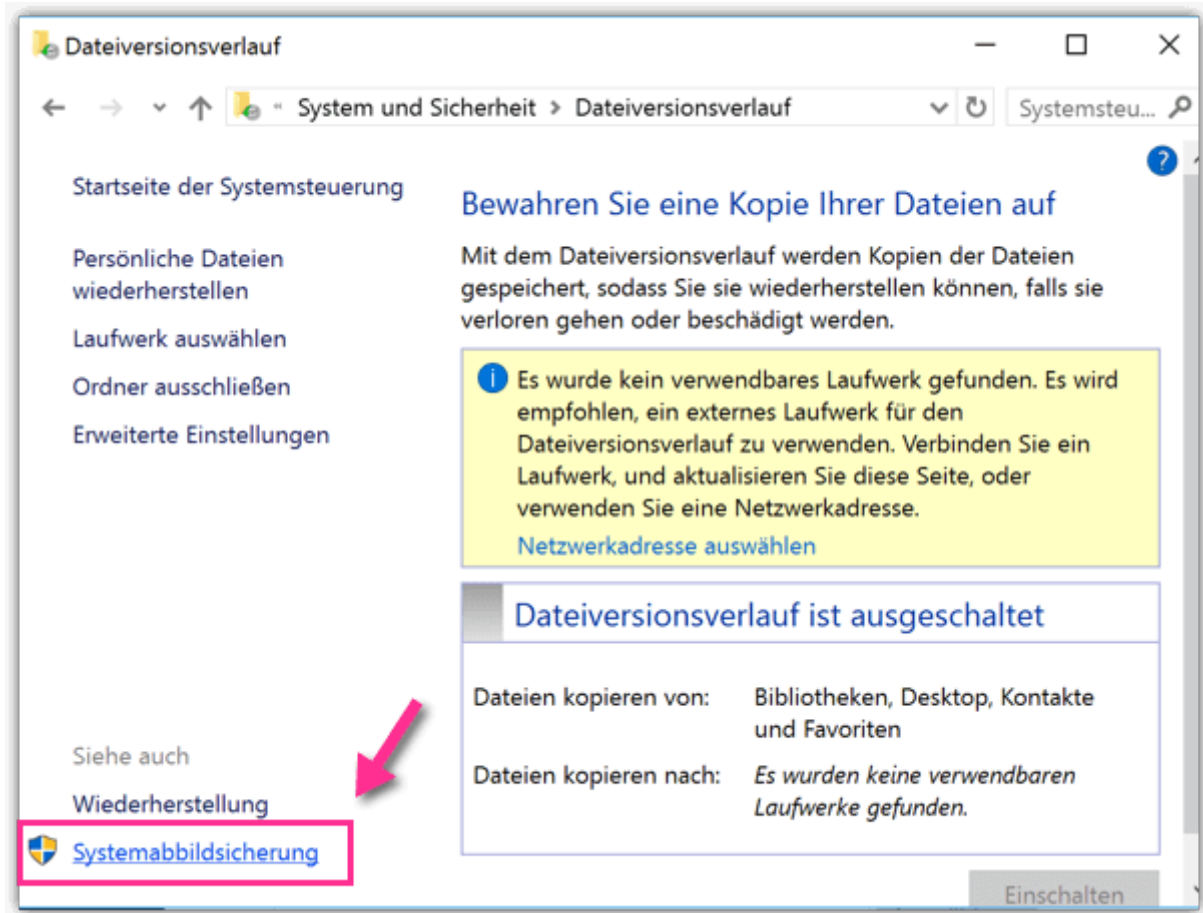


3) Komplettes Systemabbild mit Bordmitteln erstellen

Wollt ihr aber euer komplettes System absichern und ein Windows 10-Systemabbild erstellen, so sind folgende Schritte notwendig. Damit kann man nicht nur einzelne Dateien sondern auch das komplette Betriebssystem im Fehlerfall wiederherstellen.

Achtung!!!: Speichert das Backup nie auf das Systemlaufwerk, denn im Falle eines Festplattendefekts hilft das beste Backup nichts!! Als Speicherort eignet sich am besten eine externe Festplatte oder ein Serverlaufwerk mit genügend Speicherplatz.

- Navigiert, wie oben beschrieben, erneut in das „Weitere Optionen“-Menü.
- Scrollt in den Sicherungsoptionen bis an das Ende der Seite und klickt auf „Siehe erweiterte Einstellungen“.
- Wählt im neu geöffneten Fenster unten links „Systemabbildsicherung“ und anschließend „Systemabbild erstellen“ aus.
- Wählt die Festplatte aus, auf dem das Systemabbild erstellt werden soll und bestätigt mit „Weiter“.
- Windows zeigt euch an, welche Laufwerke mit dem Backup gesichert werden. Schließt den Vorgang mit „Sicherung starten“ ab



4) Alternative Backup-Methoden

Neben den Windows-Bordmitteln könnt ihr für eure Datensicherung auch auf verschiedene Software-Lösungen zurückgreifen.

- [Acronis](#)
- [Duplicati](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_09:05_09_04

Last update: **2019/05/14 15:55**



5.10) LINUX

Linux ist ein freies Multiplattform-Mehrbenutzer-Betriebssystem, das den Linux-Kernel enthält. Im praktischen Einsatz werden meist sogenannte Linux-Distributionen genutzt, in denen der Linux-Kernel und verschiedene Software zu einem fertigen Paket zusammengestellt sind.

Das wichtigste gleich vorweg, in Linux ist alles eine Datei -> Everything is a file.

Everything is a file beschreibt eine der definierenden Eigenschaften von Unix und seinen Abkömmlingen, demnach Ein-/Ausgabe-Ressourcen wie Dateien, Verzeichnisse, Geräte (z. B. Festplatten, Tastaturen, Drucker) und sogar Interprozess- und Netzwerk-Verbindungen als einfache Byteströme via Dateisystem verfügbar sind

- [5.10.1\) Aufbau des Betriebssystems](#)
- [5.10.2\) Benutzer](#)
- [5.10.3\) Dateimanagement](#)
- [5.10.4\) Dateirechte](#)
- [5.10.5\) Inodes](#)
- [5.10.6\) Distributionen und Desktops](#)
- [5.10.7\) Konsole/Bash/Terminal](#)
- [5.10.8\) Shell Scripts](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10



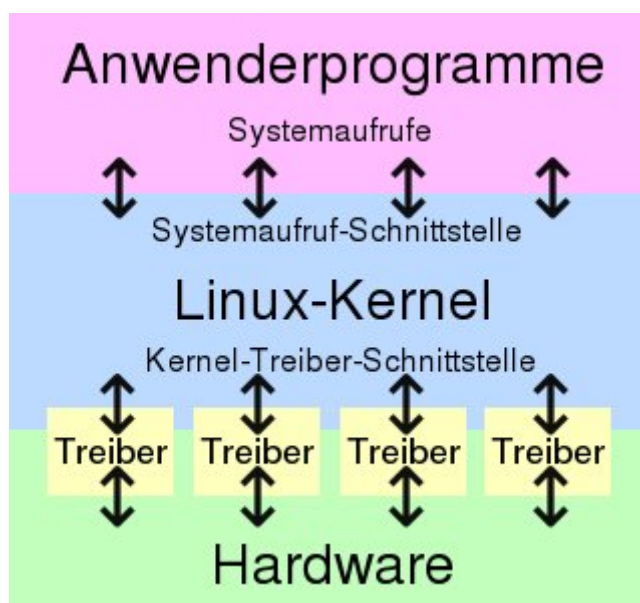
Last update: **2019/05/14 15:59**

5.10.1) Aufbau des Betriebssystems

Das zentrale Kernstück des Betriebssystems, der Linux-Kernel (meist nur Kernel genannt) bildet eine Trennschicht zwischen Hardware und Anwenderprogrammen. Das heißt, wenn ein Programm auf ein Stück Hardware zugreifen will, so kann es niemals direkt darauf zugreifen, sondern nur über das Betriebssystem.

Dazu bedient sich das Programm der **Systemaufrufe**. Über den Systemaufruf teilt das Anwenderprogramm dem Betriebssystem mit, dass es etwas zu tun gibt. Will etwa ein Programm eine Zeile Text auf dem Bildschirm ausgeben, so wird ein Systemaufruf gestartet, dem der Text übergeben wird. Das Betriebssystem erst schreibt ihn auf den Bildschirm.

Auf der anderen Seite muss das Betriebssystem die Möglichkeit haben, mit den einzelnen Hardware-Komponenten zu sprechen. Mittels seiner **Treiberschnittstelle** spricht es spezielle Geräte-Treiber an. Erst die Treiber kommunizieren dann direkt mit den Geräten.



Zu den **Anwenderprogrammen** zählen alle von uns gestarteten Programme (Videoplayer, Webbrowser ...), wie auch die grafische Oberfläche des Betriebssystems, das Desktop-Environment. Letzteres ist nicht ein Programm, sondern eine Sammlung von Programmen, die zusammen die gewohnten Funktionalitäten beisteuern.

Ein ganz spezielles Anwenderprogramm ist die Shell - die „Benutzeroberfläche“. Es existieren viele verschiedene Shells - wir werden hier mit der Bash (Bourne again shell) arbeiten. Diese ist die Standardshell auf Linuxsystemen. Alle Shells stellen dem Benutzer eine Kommandozeile zur Verfügung, mit der Befehle eingegeben werden können, die direkt als Systemaufrufe an das Betriebssystem weitergeleitet werden.

Linux ist ein **Multitasking-Betriebssystem**: das heißt, es können mehrere Prozesse - so nennt man Programme, sobald sie in den Speicher geladen sind und laufen - gleichzeitig laufen. Das bedingt, dass das System die verfügbare Rechenzeit des Prozessors in kleine Zeitscheiben aufteilt (im Millisekundenbereich), die dann den jeweiligen Prozessen zur Verfügung stehen. Diese Aufgabe übernimmt eine übergeordnete Instanz - der Scheduler. Dieser verwaltet die Zuteilung der Zeitscheiben an die verschiedenen Prozesse.

Daher kann kein Prozess die ganze Rechenleistung für sich beanspruchen und auch ein „hängender“ Prozess kann nicht das ganze System lahmlegen.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_01



Last update: **2019/05/14 15:58**

5.10.2) Benutzer

Linux ist auch ein **Multius**er-System, das heißt, es können mehrere Benutzer an verschiedenen Terminals auf dem selben Rechner arbeiten. Dazu ist es natürlich notwendig, dass jeder Benutzer eindeutig identifiziert ist. Die User (engl., Benutzer) werden zwar mit ihren Namen verwaltet, intern arbeitet ein Unix-System aber mit Usernummern. Jeder Benutzer hat also eine Nummer welche UserID oder kurz UID genannt wird. Jeder Benutzer ist auch Mitglied mindestens einer Gruppe. Es kann beliebig viele Gruppen in einem System geben und auch sie haben intern Nummern (GroupID oder GID). Im Prinzip sind Gruppen nur eine Möglichkeit, noch detailliertere Einstellungsmöglichkeiten zu haben, wer was darf.

Eine spezielle Rolle hat der Benutzer mit der UserID 0 - er ist Root (engl., Wurzel). Root steht außerhalb aller Sicherheitseinrichtungen des Systems - kurz - er darf alles. Er kann mit einem Befehl das ganze System zerstören, er kann die Arbeit von Wochen und Monaten löschen usw. Aus diesem Grund meldet sich auch der Systemverwalter im Normalfall als normaler Benutzer an - zum Root-Benutzer wird er nur dann, wenn er Systemverwaltungsarbeiten abwickelt, die diese Identität benötigen.

Benutzertypen

root

Der Benutzer root ist mit allen Rechten ausgestattet, die ihm die Administration (bei Unachtsamkeit natürlich auch die Beschädigung!) des Systems erlauben. Diesem auch als Superuser bezeichneten Benutzer ist immer die UID 0 zugeordnet.

Systembenutzer

Je nach System kann eine Vielzahl von Prozessen und Diensten erwünscht sein, die bereits beim Hochfahren des Systems verfügbar sein sollen. Nicht jeder dieser Prozesse benötigt jedoch die volle Rechteeinstattung des Superusers. Man möchte natürlich so wenige Prozesse wie nur möglich unter einer root Kennung starten, da die weitreichenden Rechte solcher Prozesse unnötige Möglichkeiten für Missbrauch und Beschädigung des Systems liefern.

Ein Systembenutzerkonto ist in diesem Sinne ein Benutzerkonto, das jedoch (nahezu) ausschließlich zur Ausführung von Programmen unter einer speziellen Benutzerkennung verwendet wird. Kein menschlicher Benutzer meldet sich normalerweise unter einem solchen Konto an.

Standardbenutzerkonto

Dies ist das normale Benutzerkonto, unter welchem jeder üblicherweise arbeiten sollte.

Die zentralen Benutzerdateien

Die Dateien zur Benutzerverwaltung finden Sie unter Linux im Verzeichnis `/etc`. Es handelt sich dabei um die Dateien **`/etc/passwd`**, **`/etc/shadow`** und **`/etc/group`**.

`/etc/passwd`

Die Datei `/etc/passwd` ist die zentrale Benutzerdatenbank.

Mit `cat /etc/passwd` können Sie einen Blick in diese zentrale Benutzerdatei werfen. Hier werden alle Benutzer des Systems aufgelistet. Zu beachten ist, dass alle Benutzertypen eingetragen sind, also sowohl der Superuser `root` als auch die Standard- und Systembenutzer.

Ein Benutzerkonto in der Datei `/etc/passwd` hat generell folgende Syntax:

Benutzername : Passwort : UID : GID : Info : Heimatverzeichnis : Shell

Spalte	Erklärung
Benutzername	Dies ist der Benutzername in druckbare Zeichen, meistens in Kleinbuchstaben.
Passwort	Hier steht verschlüsselt das Passwort des Benutzers (bei alten Systemen). Meist finden Sie dort ein <code>x</code> . Dies bedeutet, dass das Passwort verschlüsselt in der Datei <code>/etc/shadow</code> steht. Es ist auch möglich, den Eintrag leer zu lassen. Dann erfolgt die Anmeldung ohne Passwortabfrage (in der Datei <code>/etc/shadow</code> muss dann an Stelle des verschlüsselten Passwortes ein <code>*</code> stehen).
UID	Die Benutzer-ID des Benutzers. Die Zahl hier sollte größer als 100 sein, weil die Zahlen unter 100 für Systembenutzer vorgesehen sind. Weiterhin muss die Zahl aus technischen Gründen kleiner als 64000 sein.
GID	Die Gruppen-ID des Benutzers. Auch hier muss die Zahl wie bei der UID kleiner als 64000 sein.
Info	Hier kann weitere Information vermerkt werden, wie z.B. der vollständige Name des Benutzers und persönliche Angaben (Telefonnummer, Abteilung, Gruppenzugehörigkeit u.ä.).
Heimatverzeichnis	Das Heimatverzeichnis des Benutzers bzw. das Startverzeichnis nach dem Login.
Shell	Die Shell, die nach der Anmeldung gestartet werden soll. Bleibt dieses Feld frei, dann wird die Standardshell <code>/bin/sh</code> gestartet.

Hier ein Beispiel für einen Systembenutzer:

`uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash`

Der Benutzer heißt `uucp`, das Passwort ist in der Datei `/etc/shadow` gespeichert (`x`), die UID ist 10, die GID 14, als erläuternde Bezeichnung trägt der Benutzer den Namen „Unix-to-Unix CoPy system“, das Startverzeichnis nach der Anmeldung ist `/etc/uucp`, und die vorgeschlagene Shell ist die `bash`.

An dieser Stelle sei nochmals darauf hingewiesen, dass die meisten Linux-Distributionen komfortable Werkzeuge zur Benutzerverwaltung mitliefern und es auch eine Reihe von Befehlen gibt, die für die Benutzerverwaltung verwendet werden können

/etc/shadow

Bei früheren Versionen von Linux speicherte man die die Passwörter direkt in die passwd-Datei. Allerdings war dies durch einen sogenannten Wörterbuchangriff und der beispielsweise mit Hilfe des Programmes crypt möglich, diese Passwörter in vielen Fällen zu entschlüsseln und auszulesen. Deshalb hat man die Datei /etc/shadow eingeführt, in der die Angaben über die Passwörter durch ein spezielles System besser geschützt werden.

Der Eintrag in diese Datei erfolgt nach einem ähnlichen Schema, wie in der Datei /etc/passwd:

Benutzername : Passwort : DOC : MinD : MaxD : Warn : Exp : Dis : Res

Benutzername	Dies ist der Benutzername in druckbaren Zeichen, meistens in Kleinbuchstaben.
Passwort	Hier steht verschlüsselt das Passwort des Benutzers. Wenn hier ein * oder ! steht, dann bedeutet dies, dass kein Passwort vorhanden bzw. eingetragen ist.
DOC	Day of last change: der Tag, an dem das Passwort zuletzt geändert wurde. Besonderheit hier: Der Tag wird als Integer-Zahl in Tagen seit dem 1.1.1970 angegeben.
MinD	Minimale Anzahl der Tage, die das Passwort gültig ist.
MaxD	Maximale Anzahl der Tage, die das Passwort gültig ist.
Warn	Die Anzahl der Tage vor Ablauf der Lebensdauer, ab der vor dem Verfall des Passwortes zu warnen ist.
Exp	Hier wird festgelegt, wieviele Tage das Passwort trotz Ablauf der MaxD noch gültig ist.
Dis	Bis zu diesem Tag (auch hier wird ab dem 1.1.1970 gezählt) ist das Benutzerkonto gesperrt
Res	Reserve, dieses Feld hat momentan keine Bedeutung.

Es folgt wieder ein Beispiel:

```
selflinux:/heSIGnYDr6MI:11995:1:99999:14:::
```

Der Benutzer heißt selflinux, das Passwort lautet verschlüsselt „/heSIGnYDr6MI“. Es wurde zuletzt geändert, als 11995 Tage seit dem 1.1.1970 vergangen waren. Das Passwort ist minimal einen Tag gültig, maximal 99999 Tage (was man als immer deuten kann - 99999 Tage sind ca. 274 Jahre). Es soll ab 14 Tage vor Ablauf des Passwortes gewarnt werden. Die anderen Werte sind vom Administrator nicht definiert und bleiben daher leer.

/etc/group

In dieser Datei finden Sie die Benutzergruppen und ihre Mitglieder. In der Datei /etc/passwd wird mit der GID eigentlich schon eine Standardgruppe für den Benutzer festgelegt. In der /etc/group können Sie weitere Gruppenzugehörigkeiten definieren. Das hat in der Praxis vor allem in Netzwerken eine große Bedeutung, weil Sie so in der Lage sind, z.B. Gruppen für Projekte oder Verwaltungseinheiten zu bilden. Für diese Gruppen kann man dann entsprechend die Zugriffsrechte einstellen. Dies hat dann wiederum den Vorteil, dass man die Daten gegen eine unbefugte Benutzung absichern kann.

Der Eintrag einer Gruppe in die Datei sieht so aus:

Gruppenname : Passwort : GID : Benutzernamen (Mitgliederliste)

Gruppenname	Der Name der Gruppe in druckbare Zeichen, auch hier meistens Kleinbuchstaben.
Passwort	Die Besonderheit hier ist folgende: Wenn das Passwort eingerichtet ist, können auch Nichtmitglieder der Gruppe Zugang zu den Daten der Gruppe erhalten, wenn ihnen das Passwort bekannt ist. Ein x sagt hier aus, dass das Passwort in /etc/gshadow abgelegt ist. Der Eintrag kann auch entfallen, dann ist die Gruppe nicht durch ein Passwort geschützt. In diesem Fall kann jedoch auch kein Benutzer in die Gruppe wechseln, der nicht in diese Gruppe eingetragen ist.
GID	Gruppen-ID der Gruppe
Benutzernamen	hier werden die Mitglieder der Gruppe eingetragen. Diese sind durch ein einfaches Komma getrennt.

Für einen korrekten Eintrag in die /etc/group reicht eigentlich der Gruppenname und die GID aus. Damit ist die Gruppe dem System bekannt gemacht. Die Felder für das Passwort und die Benutzernamen können frei bleiben.

Soll der Benutzer nur in seiner Standardgruppe bleiben, ist kein Eintrag in die /etc/group notwendig. Hier reicht der Eintrag in die /etc/passwd völlig aus, weil dort die Standardgruppe schon mit angegeben wird. Nur wenn der Benutzer in weiteren bzw. mehreren Gruppen Mitglied sein soll, muss dies in die /etc/group-Datei eingetragen werden. Für Passwörter gilt das oben in der Tabelle Gesagte.

Hier sehen Sie ein Beispiel für einen Eintrag:

```
dialout:x:16:root,tatiana,steuer,selflinux
```

Sie sehen eine Gruppe mit der GID „16“ und den Namen dialout. (Zur Information: dialout erlaubt es normalen Benutzern einen ppp-Verbindungsaufbau zu starten, normalerweise hat nur root dieses Recht). Das x bedeutet hier, dass das Passwort in /etc/shadow abgelegt ist. Da in /etc/gshadow hier bei Passwort ein * steht, ist also kein Passwort für die Gruppe vorhanden (Das bedeutet wiederum, dass nur die eingetragenen Mitglieder Zugang zu dieser Gruppe haben). Mitglieder der Gruppe sind: root, tatiana, steuer, selflinux.

Benutzerklassen: user, group und others

Aus der Sicht des Systems existieren drei Benutzerklassen, wenn entschieden werden soll, ob die Berechtigung für einen Dateizugriff existiert oder nicht. Soll beispielsweise eine Datei gelöscht werden, so muss das System ermitteln, ob der Benutzer, welcher die Datei löschen möchte, das erforderliche Recht besitzt:

```
user@linux ~$ rm testdatei rm: Entfernen (unlink) von „testdatei“ nicht möglich: Keine Berechtigung
```

In diesem Fall wurde dem rm Kommando der beabsichtigte löschende Zugriff auf die Datei verwehrt - der ausführende Benutzer hatte nicht das Recht, die Datei zu löschen. Um diese Entscheidung zu treffen, verwendet das System das Konzept der Benutzerklassen. Drei Benutzerklassen werden unterschieden: user, group und others. Jede dieser Benutzerklassen ist wiederum in ein Lese-, Schreib- und Ausführrecht unterteilt. Diese werden im Folgenden als Berechtigungsklassen bezeichnet. Somit ergibt sich folgende Körnung für die einfachen Zugriffsrechte einer Datei:

Recht	Beschreibung
u ser-read	Leserecht für Dateieigentümer
u ser-write	Schreibrecht für Dateieigentümer
u ser-execute	Ausführrecht für Dateieigentümer
g roup-read	Leserecht für Gruppe des Dateieigentümers
g roup-write	Schreibrecht für Gruppe des Dateieigentümers
g roup-execute	Ausführrecht für Gruppe des Dateieigentümers
o ther-read	Leserecht für alle anderen Benutzer
o ther-write	Schreibrecht für alle anderen Benutzer
o ther-execute	Ausführrecht für alle anderen Benutzer

Benutzerklassen sind also eng mit der Eigentümerschaft von Dateien verbunden. Jede Datei und jedes Verzeichnis ist sowohl einem Benutzer (einer UID) als auch einer Gruppe (einer GID) zugeordnet. UID und GID gehören zur elementaren Verwaltungsinformation von Dateien und Verzeichnissen und werden in der sogenannten Inode gespeichert.

Beim Zugriff auf eine Datei werden nun UID und GID des zugreifenden Prozesses mit UID und GID der Datei verglichen. Ist other-read gesetzt, darf jeder Benutzer lesend zugreifen und ein weiterer Vergleich erübrigt sich. Ist lediglich group-read gesetzt, muss der Zugreifende mindestens der Gruppe des Dateieigentümers angehören, d.h. eine identische GID aufweisen. Ist ausschließlich user-read gesetzt, so darf nur der Eigentümer selbst die Datei lesen. root ist von dieser Einschränkung freilich ausgenommen. („Ich bin root, ich darf das!“). Von sehr speziellen Ausnahmen abgesehen, die sich außerhalb der hier besprochenen Rechteklassen bewegen, ist root in seinen Aktionen in keinerlei Weise eingeschränkt.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_02



Last update: **2019/05/14 15:58**

5.10.3) Dateimanagement

Nachdem unter Linux das Prinzip **Everything is a file** gilt, werden hier die Besonderheiten von Dateien beschrieben.

Dateien

Datei- und Verzeichnisnamen können bis zu 255 Zeichen lang sein. Dabei wird in jedem Fall zwischen Groß- und Kleinschreibung unterschieden. Die Dateinamen

- DATEI
- datei
- Datei

bezeichnen drei unterschiedliche Dateien. Ein Dateiname darf beliebig viele Punkte enthalten, also zum Beispiel auch Datei.Teil.1.txt. Ein Punkt gilt als normales Zeichen in einem Dateinamen. Dateien, die mit Punkt beginnen, gelten als versteckt und werden normalerweise nicht angezeigt - zum Beispiel .datei. Das Zeichen zum Trennen von Verzeichnis- und Dateinamen ist der Slash („/“) statt dem Backslash („\") bei Windows.

Es gibt verschiedene Dateiarten: (in Klammer die offizielle Darstellung, wie sie symbolisiert werden)

- Normale Dateien (-)
- Verzeichnisse (d)
- Symbolische Links (l)
- Blockorientierte Geräte (b)
- Zeichenorientierte Geräte (c)
- Named Pipes (p)

Wir sehen hier schon, dass auch Verzeichnisse bloß eine bestimmte Dateiart sind. Eine spezielle nämlich, in der andere Dateien aufgelistet sind. Mit einem Dateibrowser (von Windows kennen wir „Explorer“, bei Apple den „Finder“) sehen wir uns immer nur genau diese Verzeichnisse an, sofern wir nicht mittels verschiedener Plugins die Dateien selbst auswerten und Textdokumente, Bilder anzeigen oder Videos und Musik wiedergeben.

In einem Unix-Dateisystem hat jede einzelne Datei jeweils einen Eigentümer und eine Gruppenzugehörigkeit. Neben diesen beiden Angaben besitzt jede Datei noch einen Satz Attribute, die bestimmen, wer die Datei wie benutzen darf. Diese Attribute werden dargestellt als „rwx“. Dabei steht r für lesen (read), w für schreiben (write) und x für ausführen (execute).

Das Dateisystem

Das Dateisystem ist die Ablageorganisation auf einem Datenträger eines Computers. Um die Funktionsweise zu verstehen, betrachten wir einen Datenträger, die Festplatte, näher:

Die Festplatte besteht aus mehreren Scheiben mit einer magnetisierbaren Oberfläche, auf die die

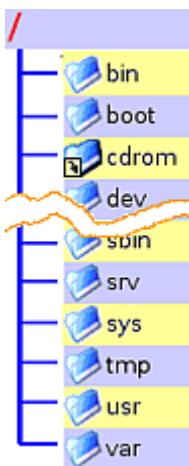
Schreibköpfe unsere Daten als Einsen (ein) und Nullen (aus) abspeichern. Um diese aber vernünftig adressieren zu können, benutzen wir Dateisysteme. Ein solches teilt die Festplatte (eigentlich die „Partition“, denn die Festplatte wird häufig in mehrere Partitionen aufgeteilt, die dann unabhängig formatiert werden können) in kleine Einheiten, die „Blöcke“, welche aus Performancegründen häufig noch zu „Clustern“ zusammengefasst werden.

Der Block (oder Cluster) ist dann die kleinste Einheit, in die eine Datei geschrieben wird, jede Datei benötigt dadurch immer diesen Speicherplatz (oder ein vielfaches) auf der Festplatte.

Von Windows kennen wir NTFS und FAT32, bzw. Apple-Benutzer werden schon von HFS+ gehört haben. Unter Linux werden meist ext2, ext3 oder ext4 (Second, Third bzw. Fourth Extended File System) verwendet. „ext3“ unterscheidet sich von „ext2“ nur dadurch, dass zusätzlich ein „Journal“ geschrieben wird, welches bei Systemabstürzen eine zuverlässige Wiederherstellung möglich macht. „ext4“ ist eine performantere Weiterentwicklung von „ext3“ und heute Standard. Daneben gibt es gelegentlich noch ReiserFS, XFS oder JFS, aber die Wahl des Dateisystems bestimmt tatsächlich immer das Abwägen zwischen höherer Sicherheit und schnellerer Schreibgeschwindigkeit - mit oder ohne Journal.

Verzeichnisstruktur

Das Dateisystem beginnt mit einem Wurzelverzeichnis (auch Rootverzeichnis genannt - /). Es enthält im Regelfall keine Dateien, sondern nur die folgenden Verzeichnisse (Ubuntu):



/bin

Von: binaries (Programme); muss bei Systemstart vorhanden sein; enthält für Linux unverzichtbare Programme; diese Programme können im Gegensatz zu /sbin von allen Benutzern ausgeführt werden; /bin darf keine Unterverzeichnisse enthalten.

/boot

Muss bei Systemstart vorhanden sein; Enthält zum Booten benötigte Dateien.

/dev

Von devices (Geräte); muss bei Systemstart vorhanden sein; enthält alle Gerätedateien, über die die Hardware im Betrieb angesprochen wird

/etc

Von: et cetera („alles übrige“), später auch: editable text configuration (änderbare Text Konfiguration); muss bei Systemstart vorhanden sein; enthält Konfigurations- und Informationsdateien des Basissystems.

- /etc/init.d: dort liegen alle Start- und Stopskripte
- /etc/opt: Verzeichnisse und Konfigurationsdateien für Programme in /opt
- /etc/network: Verzeichnisse und Konfigurationsdateien des Netzwerkes
-

/home

Von: home-directory (Heimatverzeichnis); enthält pro Benutzer ein Unterverzeichnis; jedes Verzeichnis wird nach dem Anmeldenamen benannt

/lib

Von: libraries (Bibliotheken); muss bei Systemstart vorhanden sein; enthält unverzichtbare Bibliotheken fürs Booten und die dynamisch gelinkten Programme des Basissystems;

/lost+found

(verloren und gefunden); Dateien und Dateifragmente, die beim Versuch, ein defektes Dateisystem zu reparieren, übrig geblieben sind.

/media

Für (Speicher-)Medien. Enthält Unterverzeichnisse, welche als mount- oder Einhängepunkte für transportable Medien wie z.B. externe Festplatten, USB-Sticks, CD-ROMs, DVDs und andere Datenträger dienen. Ubuntu legt hier auch die Einhängepunkte für Partitionen an. Unterverzeichnisse sind u.a.:

- /media/floppy: Einhängepunkt für Disketten
- /media/cdrom0: Einhängepunkt für CD-ROMs

/mnt

Von: mount (eingehängt); normalerweise leer; kann für temporär eingehängte Partitionen verwendet werden. Für Datenträger, die hier eingehängt werden, wird im Gegensatz zu /media kein Link auf dem Desktop angelegt

/opt

Von: optional; ist für die manuelle Installation von Programmen gedacht, die ihre eigenen Bibliotheken mitbringen und nicht zur Distribution gehören;

/proc

Von: processes (laufende Programme); muss bei Systemstart vorhanden sein; enthält Schnittstellen zum aktuell geladenen Kernel und seinen Prozeduren; Dateien lassen sich mittels cat auslesen; Beispiele: version (Kernelversion), swaps (Swapspeicherinformationen), cpuinfo, interrupts, usw.;

/root

Ist das Homeverzeichnis des Superusers (root). Der Grund, wieso sich das /root-Verzeichnis im Wurzelverzeichnis und nicht im Verzeichnis /home befindet, ist, dass das Homeverzeichnis von Root immer erreichbar sein muss, selbst wenn die Home-Partition aus irgendeinem Grund (Rettungs-Modus, Wartungsarbeiten) mal nicht eingehängt ist.

/sbin

Von: system binaries (Systemprogramme); muss bei Systemstart vorhanden sein; enthält alle Programme für essentielle Aufgaben der Systemverwaltung; Programme können nur vom Systemadministrator (root) oder mit Superuserrechten ausgeführt werden

/srv

Von: services (Dienste); Verzeichnisstruktur noch nicht genau spezifiziert; soll Daten der Dienste enthalten; unter Ubuntu in der Regel leer

/sys

Von: system; im FHS noch nicht spezifiziert; erst ab Kernel 2.6. im Verzeichnisbaum enthalten; besteht ebenso wie /proc hauptsächlich aus Kernelschnittstellen

/tmp

Von: temporary (temporär); enthält temporäre Dateien von Programmen; Verzeichnis soll laut FHS beim Booten geleert werden.

/usr

Von: user (siehe: Herkunft); enthält die meisten Systemtools, Bibliotheken und installierten Programme; der Name ist historisch bedingt - früher, als es /home noch nicht gab, befanden sich hier auch die Benutzerverzeichnisse;

- /usr/bin : Anwenderprogramme; Hier liegen die Desktopumgebungen und die dazu gehörigen Programme, aber auch im Nachhinein über die Paketverwaltung installierte Programme, wie Audacity

/var

Von variable (variabel); enthält nur Verzeichnisse; Dateien in den Verzeichnissen werden von den Programmen je nach Bedarf geändert (im Gegensatz zu /etc); Beispiele: Log-Dateien, Spielstände, Druckerwarteschlange

- /var/log : Alle Log-Dateien der Systemprogramme; Beispiele: Xorg.0.log (Log-Datei des XServer), kern.log (Logdatei des Kernels), dmesg (letzte Kernelmeldungen), messages (Systemmeldungen); Siehe auch Logdateien

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_03



Last update: **2019/05/14 15:59**

5.10.4) Dateirechte

UNIX-Systeme wie Linux verwalten ihre Dateien in einem virtuellen Dateisystem (VFS, Virtual File System). Dieses ordnet jeder Datei über eindeutig identifizierbare Inodes unter anderem folgende Eigenschaften zu:

- Dateityp (einfache Datei, Verzeichnis, Link, ...)
- Zugriffsrechte (Eigentümer-, Gruppen- und sonstige Rechte)
- Größe
- Zeitstempel
- Verweis auf Dateinhalt

Jedes unter Linux gängige UNIX-Dateisystem (z.B. ext2/3/4, ReiserFS, xfs usw.) unterstützt diese Rechte. Gar nicht umgesetzt werden die Rechte hingegen auf VFAT-Dateisystemen; dort können Dateirechte lediglich beim Einhängen simuliert werden. Partitionen mit dem Windows-Dateisystem NTFS werden zwar in Linux standardmäßig ähnlich wie VFAT-Partitionen behandelt; mit den Mount-Optionen `permissions` und `acl` lässt sich aber auch auf NTFS-Partitionen eine echte Rechteverwaltung wie bei UNIX-Dateisystemen einrichten. Siehe hierzu Windows-Partitionen einbinden sowie NTFS-3G.

Rechte in symbolischer Darstellung

Im Terminal lassen sich die Rechte mit dem Befehl `ls -l` anzeigen. Im Folgenden sind als Beispiel die Dateirechte des Verzeichnisses `/var/mail/` dargestellt

```
ls -l /var/mail/  
drwxrwsr-x 2 root mail 4096 Apr 23 2012 /var/mail/
```

Für die Darstellung der Rechte sind die markierten Teile der Ausgabe relevant:

- Der erste Buchstabe (d) kennzeichnet den Dateityp.
- Danach folgen die Zugriffsrechte (rwxrwsr-x).
- Eigentümer der Datei
- Gruppe

Wie auch in anderen Betriebssystemen kann man verschiedene Rechte für Eigentümer (Owner) und Gruppe (Group) vergeben. Neben Eigentümer und Gruppe gibt es noch eine weitere, allgemeine Gruppe. Diese Gruppe nennt sich andere (Others).

Darstellungsarten

Neben der symbolischen Darstellung (z.B. rwxrwxr-x) gibt es auch noch eine oktale Darstellung (nach dem Oktalsystem). Die Grundrechte (Lesen, Schreiben, Ausführen) und Kombinationen daraus werden hierbei durch eine einzelne Ziffer repräsentiert und dem Eigentümer, der Gruppe und allen anderen zugeordnet. Je nach Anwendung wird dabei von unterschiedlichen Grundwerten ausgegangen und entweder Rechte gegeben oder entzogen. Bei `chmod` wird beispielsweise von der Grundeinstellung „keine Rechte“ (000) ausgegangen und Rechte gegeben, wohingegen bei `umask` von „alle Rechte

vorhanden“ (777) ausgegangen und Rechte entzogen werden. Entsprechend sind die Werte je nach Anwendung anders.

Mögliche Werte für:

Recht(e)	chmod (octal)	umask (octal)	Symbolisch	Binäre Entsprechung
Lesen, schreiben und ausführen	7	0	rwx	111
Lesen und Schreiben	6	1	rw-	110
Lesen und Ausführen	5	2	r-x	101
Nur lesen	4	3	r-	100
Schreiben und Ausführen	3	4	-wx	011
Nur Schreiben	2	5	-w-	010
Nur Ausführen	1	6	-x	001
Keine Rechte	0	7	—	000

Hier ein paar Beispiele:

- rwxrwxrwx entspricht 0777 (chmod) oder 0000 (umask): Jeder darf lesen, schreiben und ausführen.
- rwxr-xr-x entspricht 0755 (chmod) oder 0022 (umask): Jeder darf lesen und ausführen, aber nur der Dateibesitzer darf diese Datei (oder das Verzeichnis) auch verändern.

Die nachfolgenden Erklärungen beziehen sich vor allem auf Dateien vom Typ File (ohne Kennbuchstaben) und „Ordner“ (Directory, Kennbuchstabe d).

Nach dem Dateityp kommen drei Zeichengruppen zu je drei Zeichen. Diese kennzeichnen die Zugriffsrechte für die Datei bzw. das Verzeichnis. Hat der Benutzer/Gruppe/andere ein Recht, so wird der Buchstabe dafür angezeigt; ansonsten wird ein - dafür angezeigt.

In obigen Beispiel erscheint nach dem Dateityp dann die Zeichenfolge rwxrwsr-x. Wenn man diese in drei Dreiergruppen aufteilt, erhält man diese Gruppen:

- rwx: Rechte des Eigentümers
- rws: Rechte der Gruppe
- r-x: Recht von allen anderen (others)

Die folgende Tabelle erklärt die Bedeutung der einzelnen Buchstaben, Diese stehen immer in der gleichen Reihenfolge:

Symbole für Zugriffsrechte		
Zeichen	Bedeutung	Beschreibung
r	Lesen (read) Erlaubt lesenden Zugriff auf die Datei. Bei einem Verzeichnis können damit die Namen der enthaltenen Dateien und Ordner abgerufen werden (nicht jedoch deren weitere Daten wie z.B. Berechtigungen, Besitzer, Änderungszeitpunkt, Dateiinhalte etc.).	
w	Schreiben (write) Erlaubt schreibenden Zugriff auf eine Datei. Für ein Verzeichnis gesetzt, können Dateien oder Unterverzeichnisse angelegt oder gelöscht werden, sowie die Eigenschaften der enthaltenen Dateien/Verzeichnisse verändert werden.	

Symbole für Zugriffsrechte		
Zeichen	Bedeutung	Beschreibung
x	Ausführen (execute) Erlaubt das Ausführen einer Datei, wie das Starten eines Programms. Bei einem Verzeichnis ermöglicht dieses Recht, in diesen Ordner zu wechseln und weitere Attribute zu den enthaltenen Dateien abzurufen (sofern man die Dateinamen kennt ist dies unabhängig vom Leserecht auf diesen Ordner). Statt x kann auch ein Sonderrecht angeführt sein.	

Sonderrechte

Die oben gezeigten Dateirechte kann man als Basisrechte bezeichnen. Für besondere Anwendungen gibt es zusätzlich noch besondere Dateirechte. Der Einsatz dieser ist nur dann ratsam, wenn man genau weiß, was man tut, da dies unter Umständen zu Sicherheitsproblemen führen kann.

Sonderrechte		
Zeichen	Bedeutung	Beschreibung
s	Set-UID-Recht (SUID-Bit)	Das Set-UID-Recht („Set User ID“ bzw. „Setze Benutzerkennung“) sorgt bei einer Datei mit Ausführungsrechten dafür, dass dieses Programm immer mit den Rechten des Dateibesitzers läuft. Bei Ordnern ist dieses Bit ohne Bedeutung.
s (S)	Set-GID-Recht (SGID-Bit)	Das Set-GID-Recht („Set Group ID“ bzw. „Setze Gruppenkennung“) sorgt bei einer Datei mit Ausführungsrechten dafür, dass dieses Programm immer mit den Rechten der Dateigruppe läuft. Bei einem Ordner sorgt es dafür, dass die Gruppe an Unterordner und Dateien vererbt wird, die in diesem Ordner neu erstellt werden.
t (T)	Sticky-Bit	Das Sticky-Bit („Klebrig“) hat auf modernen Systemen nur noch eine einzige Funktion: Wird es auf einen Ordner angewandt, so können darin erstellte Dateien oder Verzeichnisse nur vom Dateibesitzer gelöscht oder umbenannt werden. Verwendet wird dies z.B. für /tmp.

Die Symbole für die Sonderrechte erscheinen an der dritten Stelle der Zugriffsrechte, die normalerweise dem Zeichen x (für executable) vorbehalten ist, und ersetzen ggf. dieses. Die Set-UID/GID-Rechte werden anstelle des x für den Besitzer bzw. die Gruppe angezeigt, das Sticky-Bit anstelle des x für andere. Wenn das entsprechende Ausführrecht gesetzt ist, wird ein Kleinbuchstabe verwendet, ansonsten ein Großbuchstabe.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_04

Last update: **2019/05/14 15:59**



5.10.5) INODES

Ein Inode (englisch index node, gesprochen „eye-node“) ist die grundlegende Datenstruktur zur Verwaltung von Dateisystemen mit unixartigen Betriebssystemen. Jeder Inode wird innerhalb einer Partition eindeutig durch seine Inode-Nummer identifiziert. Jeder Namenseintrag in einem Verzeichnis verweist auf genau einen Inode. Dieser enthält die Metadaten der Datei und verweist auf die Daten der Datei beziehungsweise die Dateiliste des Verzeichnisses.

Die Anwendungssoftware unterscheidet beim Lesen oder Schreiben von Daten nicht mehr zwischen Gerätetreibern und regulären Dateien. Durch das Inode-Konzept gilt bei den Unixvarianten alles als Datei („On UNIX systems it is reasonably safe to say that everything is a file: ...“). Dadurch unterscheiden sich solche Betriebssysteme in der Verwaltung ihres Datenspeichers von anderen Systemen wie Microsoft Windows mit NTFS, aber auch von VMS oder MVS.

Grundsätzliches

Speichert man eine Datei auf einem Computer ab, so muss nicht nur der Dateiinhalt (Nutzdaten) gespeichert werden, sondern auch Metadaten, wie zum Beispiel der Zeitpunkt der Dateierstellung oder der Besitzer der Datei. Gleichzeitig muss der Computer einem Dateinamen – inklusive Dateipfad – die entsprechenden Nutzdaten und Metadaten effizient zuordnen können. Die Spezifikation, wie diese Daten organisiert und auf einem Datenträger gespeichert werden, nennt man Dateisystem. Dabei gibt es abhängig von Einsatzbereich und Betriebssystem unterschiedliche Dateisysteme. Umgesetzt wird die Dateisystemspezifikation von einem Treiber, der wahlweise als ein Kernel-Modul des Betriebssystemkern (Kernel) oder seltener als gewöhnliches Programm im Userspace umgesetzt sein kann.

Boot-block	Super-block	Inode-Liste	Datenblöcke
------------	-------------	-------------	-------------

Dateisysteme unixoider Betriebssysteme – wie Linux und macOS – verwenden sogenannte Inodes. Diese enthalten die Metadaten sowie Verweise darauf, wo Nutzdaten gespeichert sind. An einem speziellen Ort des Dateisystems, dem Superblock, wird wiederum die Größe, Anzahl und Lage der Inodes gespeichert. Die Inodes sind durchnummeriert und an einem Stück auf dem Datenträger gespeichert. Das Wurzelverzeichnis eines Dateisystems besitzt eine feste Inodennummer. Unterordner sind „gewöhnliche“ Dateien, welche eine Liste der darin enthaltenen Dateien mit der Zuordnung der dazugehörigen Inodennummern als Nutzdaten enthalten.

Soll also beispielsweise die Datei `/bin/lis` geöffnet werden, so läuft dies, vereinfacht, wie folgt ab:

- Der Dateisystemtreiber liest den Superblock aus, dadurch erfährt er die Startposition der Inodes und deren Länge, somit kann nun jeder beliebige Inode gefunden und gelesen werden.
- Nun wird der Inode des Wurzelverzeichnisses geöffnet. Da dies ein Ordner ist, befindet sich darin ein Verweis auf die Speicherstelle der Liste aller darin enthaltenen Dateien mitsamt ihren Inodennummern. Darin wird das Verzeichnis `bin` gesucht.
- Nun kann der Inode des `bin`-Verzeichnisses gelesen werden und analog zum letzten Schritt der

Inode der Datei ls gefunden werden.

- Da es sich bei der Datei ls nicht um ein Verzeichnis, sondern um eine reguläre Datei handelt, enthält ihr Inode nun einen Verweis auf die Speicherstelle der gewünschten Daten.

Aufbau

Jedem einzelnen von einem Schrägstrich / (slash) begrenzten Namen ist genau ein Inode zugeordnet. Dieser speichert folgende Metainformationen zur Datei, aber nicht den eigentlichen Namen:

- Die Art der Datei (reguläre Datei, Verzeichnis, Symbolischer Link, ...), siehe unten;
- die numerische Kennung des Eigentümers (UID, user id) und der Gruppe (GID, group id);
- die Zugriffsrechte für den Eigentümer (user), die Gruppe (group) und alle anderen (others);
- Die klassische Benutzer- und Rechteverwaltung geschieht mit den Programmen chown (change owner), chgrp (change group) und chmod (change mode). Durch Access Control Lists (ACL) wird eine feinere Rechtevergabe ermöglicht.
- verschiedene Zeitpunkte der Datei: Erstellung, Zugriff (access time, atime) und letzte Änderung (modification time, mtime);
- die Zeit der letzten Status-Änderung des Inodes (status, ctime);
- die Größe der Datei;
- den Linkzähler (siehe unten);
- einen oder mehrere Verweise auf die Blöcke, in denen die eigentlichen Daten gespeichert sind.

Reguläre Dateien

Reguläre Dateien (engl. regular files) sind sowohl Anwenderdaten als auch ausführbare Programme. Letztere sind durch das executable-Recht gekennzeichnet und werden beim Aufruf durch das System in einem eigenen Prozess gestartet. Als „ausführbar“ gelten nicht nur kompilierte Programme, sondern auch Skripte, bei denen der Shebang den zu verwendenden Interpreter angibt. Bei „dünnbesetzten Dateien“, sogenannten sparse files, unterscheidet sich die logische Größe vom durch die Datenblöcke tatsächlich belegten Festplattenplatz.

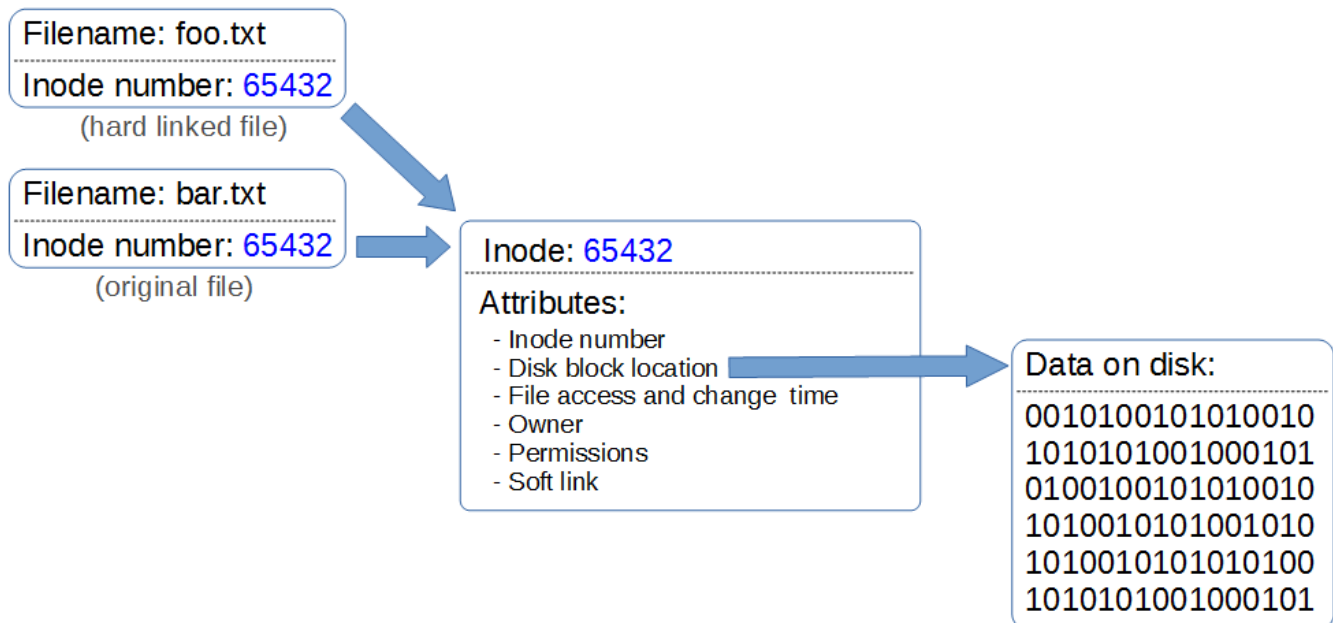
Verzeichnisse

Verzeichnisse sind Dateien, deren „Dateiinhalt“ aus einer tabellarischen Liste der darin enthaltenen Dateien besteht. Die Tabelle enthält dabei eine Spalte mit den Dateinamen und eine Spalte mit den zugehörigen Inodenummern. Bei manchen Dateisystemen umfasst die Tabelle noch weitere Informationen, so speichert ext4 darin auch den Dateityp aller enthaltenen Dateien ab, so dass dieser beim Auflisten eines Verzeichnisinhalts nicht aus den Inodes aller Dateien ausgelesen werden muss. Für jedes Verzeichnis existieren immer die Einträge . und .. als Verweise auf das aktuelle bzw. übergeordnete Verzeichnis.

Harte Links (hard links)

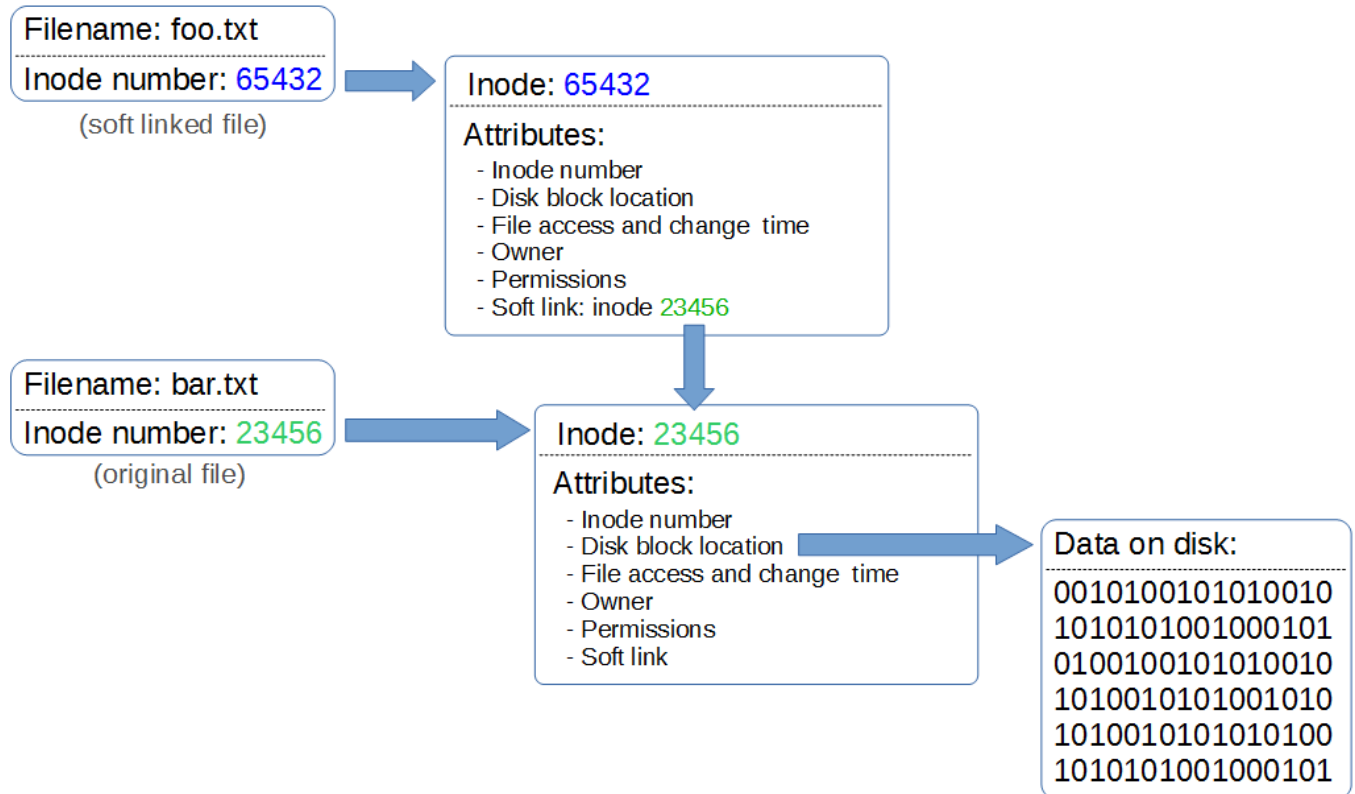
Bei Harten Links hingegen handelt es sich nicht um spezielle Dateien. Von einem Harten Link spricht man, wenn auf einen Inode mehrfach von verschiedenen Ordnern oder verschiedenen Dateinamen verwiesen wird. Alle Verweise auf den Inode sind gleichwertig, es gibt also kein Original. Im Inode gibt

der Linkzähler an, wie viele Dateinamen auf diesen verweisen, er steht nach dem Anlegen einer Datei also bei 1 und wird erhöht, sobald weitere Hardlinks für diese Datei erstellt werden. Bei einem Verzeichnis beträgt er zwei mehr, als Unterordner darin enthalten sind, da neben dem Eintrag im Ordner darüber und dem Eintrag '.' im Ordner noch die Einträge '.' in allen Unterordnern darauf verweisen. Wird eine Datei gelöscht, so wird ihr Eintrag aus dem übergeordneten Verzeichnis entfernt und der Linkzähler um eins reduziert. Beträgt der Linkzähler dann 0, wird gegebenenfalls abgewartet, bis die Datei von keinem Programm mehr geöffnet ist, und erst anschließend der Speicherplatz freigegeben.



Symbolische Links (symbolic links, soft links, symlinks)

Bei symbolischen Links handelt es sich um spezielle Dateien, die anstelle von Daten einen Dateipfad enthalten, auf den der Link verweist. Je nach Dateisystem und Länge des Dateipfads wird der Link entweder direkt im Inode gespeichert oder in einem Datenblock, auf welchen der Inode verweist.



Praxis

Die Inodennummer einer Datei lässt sich mittels des Befehls `ls -li` Dateiname anzeigen

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_05

Last update: 2019/05/14 15:59



5.10.6) LINUX-DISTRIBUTIONEN

Wie schon besprochen besteht ein Linux-Betriebssystem aus dem Linux-Kernel und einer großen Anzahl verschiedener Anwenderprogramme. Tatsächlich gibt es ein Projekt, das eine Anleitung bietet, wie man aus den Kernelquellen und selbst selektierten Programmen ein komplettes, maßgeschneidertes Betriebssystem bauen kann. Das Projekt nennt sich „Linux From Scratch“ oder „LFS“ (www.linuxfromscratch.org) und ich kann jedem, der etwas Zeit übrig hat, nur empfehlen, dies selbst einmal zu probieren. Man erhält eine ultrakompaktes, ultraschnelles Betriebssystem und kann sagen, „das ist mein eigenes reinrassiges Linux-System“. Aber was kommt dann?

Man braucht schon einige Zeit bis alle Programme, die so benötigt werden, kompiliert und konfiguriert sind und dann müssen diese auch noch laufend aktualisiert werden. Sicherheitsupdates müssen selbst organisiert und kompiliert werden. Mit all der Administrationsarbeit kommt man zu sonst nichts mehr.

Um das zu vermeiden, gibt es Linux-Distributionen. Diese bieten nicht nur einen fertigen Satz von notwendigen Programmen, sondern auch die regelmäßige Versorgung mit Updates an. Die aus meiner Sicht wichtigsten Distributionen sollen hier aufgeführt werden:

Debian



„Debian“ (www.debian.org) ist eine nicht-kommerzielle Distribution und das „Debian-Projekt“ ist nach der „Debian-Verfassung“ geregelt, die eine demokratische Organisationsstruktur vorsieht. Darüberhinaus ist das Projekt über den „Debian Social Contract“ zu völlig freier Software verpflichtet. Mit einigen 1000 Mitarbeitern ist Debian der Gigant unter den Linux-Systemen.

Da bei Debian eine „stabile Version“ immer eine wirklich stabile Version ist, sind die Entwicklungszeiten relativ lang und böse Zungen behaupten auch, dass die stabile Version schon bei Erscheinen veraltet ist.

Allerdings bietet Debian auch immer schon die zukünftigen Versionen an und so gibt es mehrere Zweige, aus denen man sich bedienen kann:

stable - wirklich stabile Version, die auch für den kommerziellen Serverbetrieb geeignet ist!

testing - die zukünftige stable-Version. Ab einem gewissen Entwicklungsstand wird die Distribution „eingefroren“ (engl. „frozen“) - d.h. es werden keine neueren Versionen von Programmen mehr aufgenommen, sondern nur noch an der Fehlerbeseitigung bei den vorhandenen gearbeitet. Dies entspricht ungefähr dem Zustand, bei dem andere Distributoren ihre „stabilen“ Versionen veröffentlichen. Da ich schon mehrmals eine testing-Version ab dem Anfangsstadium benutzt habe, glaube ich sogar sagen zu können, dass testing nie so instabil ist, wie manche andere Distribution im „ausgereiften“ Zustand. Für den Desktopbetrieb kann ich ein eingefrorenes testing jedenfalls empfehlen.

unstable - ist der erste Anlaufpunkt für neue Versionen von Paketen und Programmen, bevor sie in testing integriert werden. Man installiert sich mit unstable das neueste vom neuen, muss aber wissen,

dass das nicht immer stabil ist.

experimental - ist kein vollständiger Zweig, denn es dient nur dazu, Programme und deren Funktionen zu testen, die sonst das ganze System gefährden würden. Es enthält immer nur die gerade getesteten, bzw. die von diesen benötigten Programmpakete.

Diese Zweige haben auch immer Codenamen und sind, einer Vorliebe der frühen Entwickler folgend, immer nach Figuren aus dem Film „Toy Story“ benannt. So heißt im Moment die stable-Version „Jessie“ und „Stretch“ ist testing. unstable ist immer „Sid“, der Junge von nebenan, der die Spielsachen zerstört, aber es lässt sich auch als Abkürzung für „still in development“ (noch in Entwicklung) deuten.

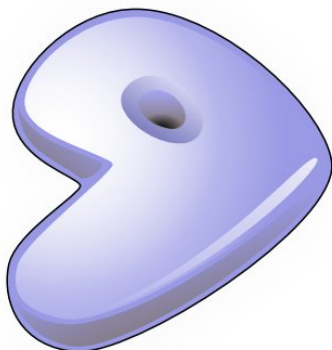
Und wer einen dieser Zweige installiert hat, kann auf eine unüberschaubare Vielzahl an Programmen zurückgreifen, auf Wunsch (und auf eigene Gefahr) auch aus den anderen Zweigen. Für Anfänger ist es wohl nur bedingt zu empfehlen, obwohl sich in den letzten Jahren sehr viel in Sachen Benutzerführung getan hat. Auch steht ein deutschsprachiges Forum (debianforum.de) zur Verfügung, wo man Hilfe bekommt und wo auch dumme Fragen gestellt werden dürfen. Für ambitionierte Linux-EinsteigerInnen, die sich auch mit den Möglichkeiten ihres Betriebssystems auseinandersetzen wollen, könnte es sogar die beste Distribution sein.

Fedora

☒ Das nicht-kommerzielle „Fedora“ (fedoraproject.org) ist der Nachfolger des traditionsreichen, kommerziellen „Red Hat Linux“, welches nicht mehr selbständig weiterentwickelt wird. Statt dessen verkauft die Firma Red Hat, das auf Fedora basierende „Red Hat Enterprise Linux“.

Fedora ist eine sehr innovative Distribution und vor allem in den USA sehr beliebt. Es werden nur völlig unter freier Lizenz stehende Inhalte akzeptiert, weshalb nach der Installation zum Beispiel keine MP3-unterstützenden Programme zu finden sind. Für Anfänger gibt es bessere Distributionen.

Gentoo Linux

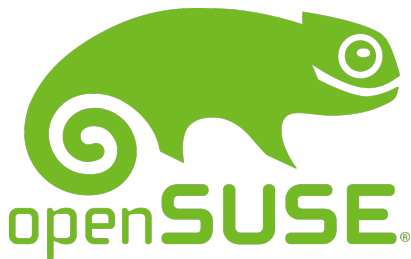


gentoo linux

Das nicht-kommerzielle „Gentoo Linux“ (www.gentoo.de) ist eine quellbasierte Linux-Metadistribution - das heißt, alle Programme, inklusive des Kernels, werden selbst kompiliert. Das klingt sehr anstrengend, ist es aber gar nicht so, da die Distribution geeignete

Werkzeuge zur Verfügung stellt, mit denen dies einfachst möglich gelingt. Auch sorgt eine große, sehr aktive „Community“ bei jedem Problem für Rat und Hilfe. Dennoch ist sie für Linux-Neulinge wohl nicht empfehlenswert.

OpenSUSE



„openSUSE“ (www.opensuse.org) ist die zweitbeliebteste Distribution am Heim-PC. Die nicht-kommerzielle Variante der von Novell aufgekauften kommerziellen SUSE-Distribution (heute „SUSE Linux Enterprise“) glänzt mit einem universellen Konfigurationswerkzeug. Sie gilt als anfängerfreundlich. Die neueste Versionsnummer 42.1 bezieht sich übrigens auf die Antwort auf die Frage „nach dem Leben, dem Universum und dem ganzen Rest“ aus Douglas Adam's „Per Anhalter durch die Galaxis“. Schon 1996 hatte die Version 4.2 diesen Bezug.

Slackware



„Slackware“ (www.slackware.com) ist die älteste noch heute existierende Distribution. Sie verzichtet aus Prinzip auf grafische Einrichtungswerkzeuge und ist daher eher nur für fortgeschrittene BenutzerInnen geeignet.

Bisher nicht vorgekommen sind Distributionen, die nur Abwandlungen anderer Distributionen sind und häufig auch deren Quellen benutzen. Vor allem von Debian gibt es unzählige davon. Sie werden als „Derivate“, oder oft auch, etwas abfällig, als „Klone“ bezeichnet. Ein solcher „Debian-Klon“ hat allerdings Geschichte geschrieben:

Ubuntu



ubuntu

Das von der Firma des Gründers gesponserte kostenlose Betriebssystem soll nach dem Willen der Entwickler ein einfach zu installierendes und leicht zu bedienendes Betriebssystem mit aufeinander abgestimmter Software sein. Es bedient sich dazu aus den Quellen von Debian unstable und hat das Ziel, nach der enormen Popularität zu schließen, eindeutig geschafft.

Tatsächlich kann Ubuntu von der Live-CD mit wenigen Mausklicks problemlos auf die Festplatte installiert werden und dann erwartet den Benutzer ein weitgehend komplettes Betriebssystem mit vielen Multimedia-Programmen. Releases erscheinen mit schöner halbjährlicher Regelmäßigkeit und der Upgrade auf diese lässt sich ebenfalls auf Mausklick bewerkstelligen. Alle 2 Jahre gibt es eine „Versionen mit verlängerter Unterstützung“ (Long Term Support oder kurz LTS), die dann deutlich stabiler als die kürzer unterstützten ist. Für EinsteigerInnen ist Ubuntu (www.ubuntu.com) bestens geeignet.

Linux Mint



Diese Distribution war ursprünglich ein Ubuntu-Klon, aber neuerdings ist Linux Mint auch als LMDE - Linux Mint Debian Edition - erhältlich. Den Entwicklern ist es wichtig, die bestmögliche Integration von Programmen zu bieten, die bei Benutzern beliebt, aber eben nicht quelloffene freie Software sind. Die anderen Distribution, inklusive Ubuntu, bieten zwar auch die Installation von „non-free“-Paketen an, aber in einem eigenen Zweig und erst nach der Basisinstallation.

Für absolute Stabilität setzt Linux Mint immer auf LTS (Ubuntu) oder stable (Debian) Versionen auf, aber die integrierten Programme erhalten auch zwischenzeitig Versionsupdates. Als Vorbild wird die Benutzerfreundlichkeit und Stabilität von Apple's OS X genannt. Auch Linux Mint (www.linuxmint.com) ist für EinsteigerInnen bestens geeignet.

GRAFISCHE OBERFLÄCHEN (GUIs, Desktops)

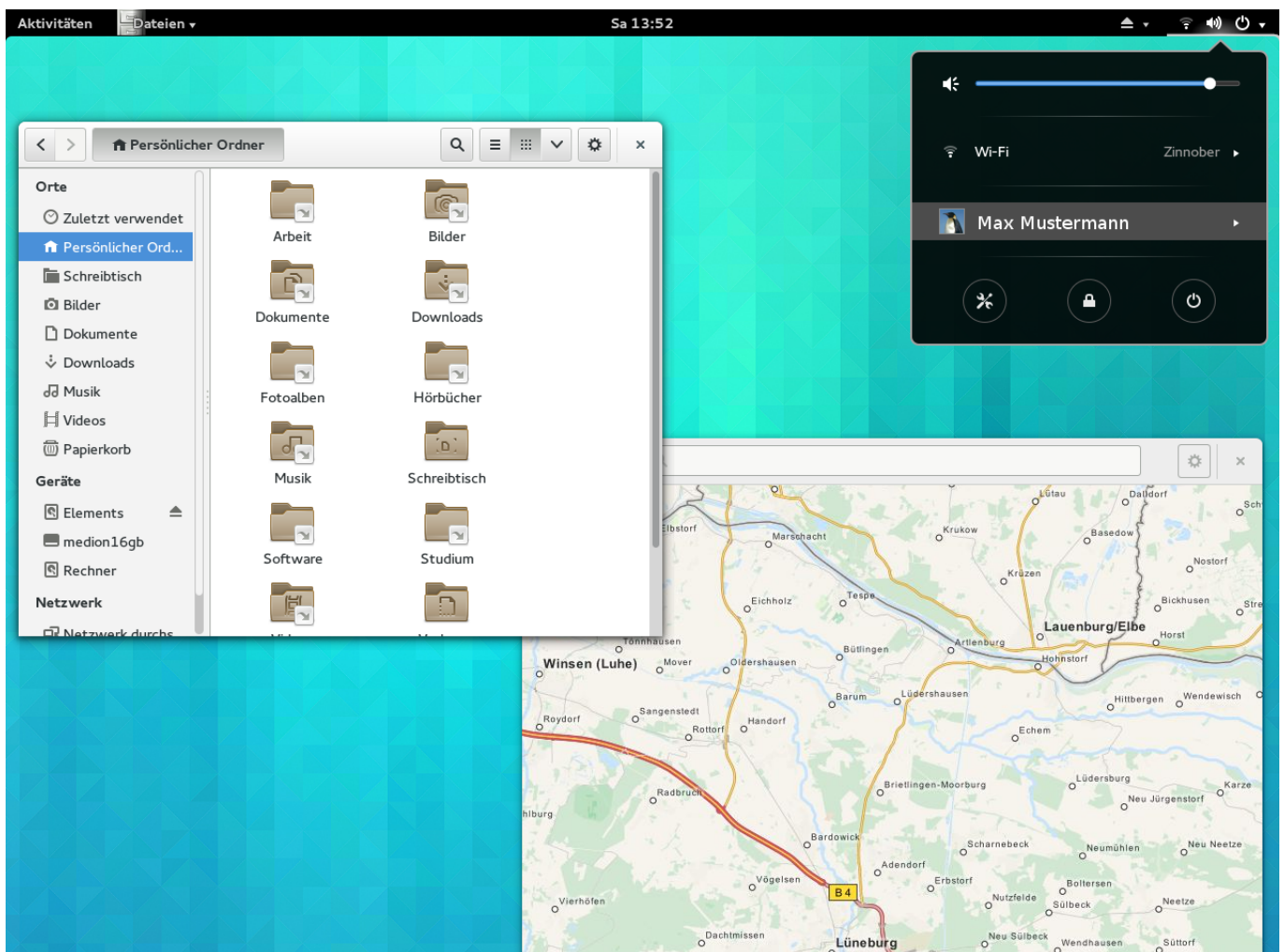
Anders als bei Microsoft und Apple gibt es bei Linux-Distributionen keinen festgelegten Desktop. Alle

Distributionen bieten die Möglichkeit den Desktop zu ändern und zunehmend kann schon bei der Installation der gewünschte Desktop festgelegt werden. Wenngleich für Anfänger im allgemeinen der Standard-Desktop der jeweiligen Distribution sicher eine gute Wahl ist, möchte ich abschließend auch noch kurz einige Desktops vorstellen. Gnome und KDE sind die Platzhirsche auf Linux-Bildschirmen.

Gnome

Gnome (Eigenschreibweise GNOME) [ɡnoʊm][3] ist eine Desktop-Umgebung für Unix- und Unix-ähnliche Systeme mit einer grafischen Benutzeroberfläche und einer Sammlung von Programmen für den täglichen Gebrauch. Gnome wird unter den freien Lizenzen GPL und LGPL veröffentlicht und ist Teil des GNU-Projekts.

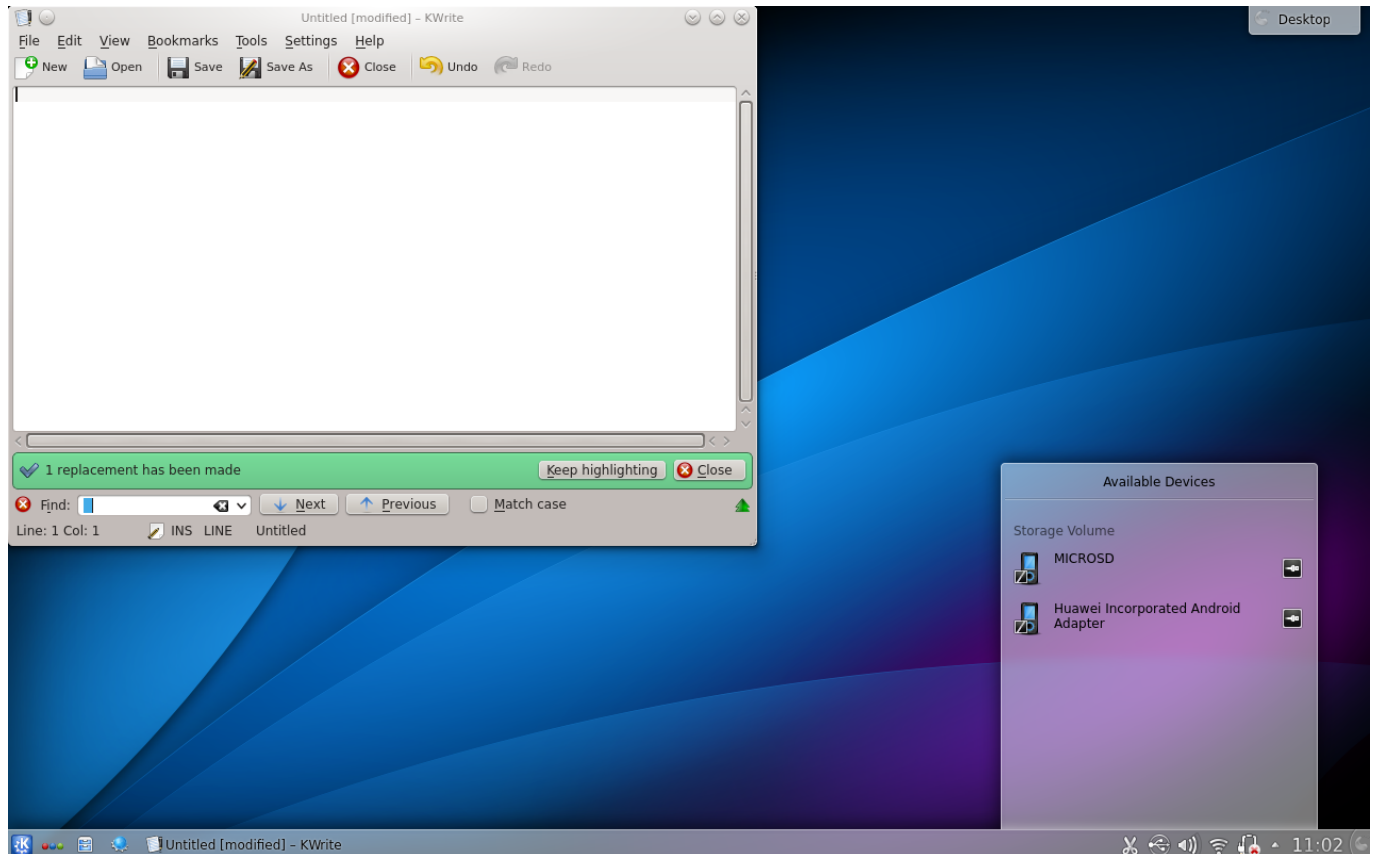
Gnome ist unter anderem der Standard-Desktop von Fedora und Ubuntu. Einige Komponenten von Gnome wurden nach Windows und MacOS portiert, etwa Evolution oder GStreamer, werden jedoch teilweise, wie im Falle von Evolution, nicht mehr länger gepflegt.



KDE

KDE ist eine Community, die sich der Entwicklung freier Software verschrieben hat. Eines der bekanntesten Projekte ist die Desktop-Umgebung KDE Plasma 5 (früher K Desktop Environment,

abgekürzt KDE).



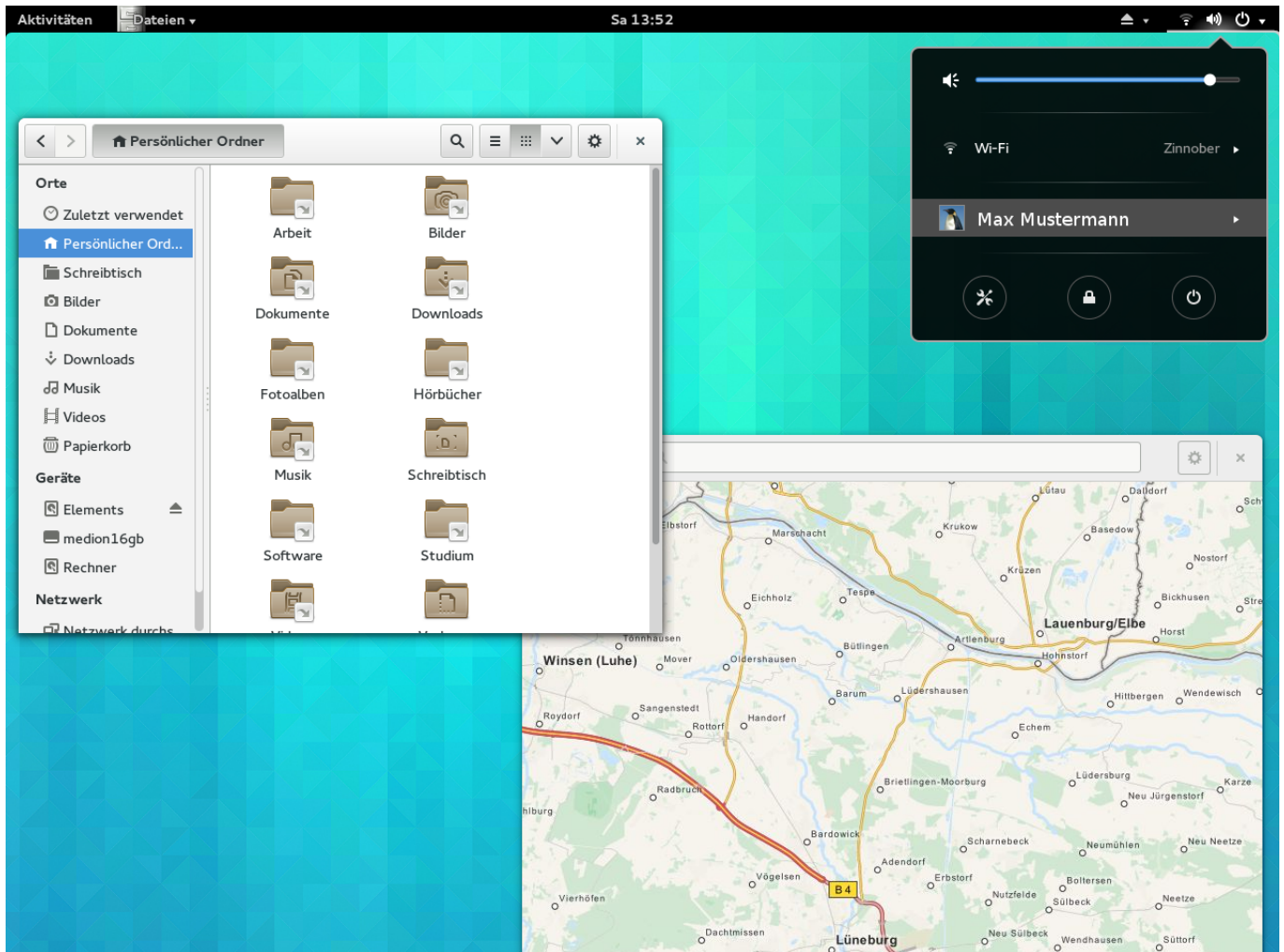
Unity

Unity ist eine durch das Unternehmen Canonical entwickelte Desktop-Umgebung für Linux-Betriebssysteme, die besonders sparsam mit Bildschirmplatz umgehen soll.



XFCE

Während erstgenannte Desktops mit Features protzen gehen die EntwicklerInnen bei XFCE einen anderen Weg. XFCE möchte einen schlanken, performanten Desktop bieten, der auf „unnötige Spielereien“ verzichtet und auch auf leistungsschwachen und zum Teil für andere Aufgaben (Multimedia) bestimmten Systemen ressourcenschonend läuft. Viele heutige Distributionen bieten XFCE als vorkonfigurierte Alternative zum Standarddesktop an.



MATE und Cinnamon

Der umstrittene Upgrade von Gnome2 auf Gnome3 führte zur Geburt von MATE. Dieser Desktop ist eine direkte Fortentwicklung von Gnome2. Cinnamon dagegen ist aus den Quellen von Gnome3 entstanden, aber deutlich performanter als dieser. Was beide gemeinsam haben - sie sind die Standarddesktops von „Linux Mint“ und nur als Ubuntu-Editionen von Linux Mint gibt es auch KDE und XFCE vorkonfiguriert.

Es können auch mehrere Desktops gleichzeitig installiert werden. Bei der Anmeldung kann dann zwischen den Desktops gewechselt werden. So kann jeder seinen Lieblingsdesktop herausfinden.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_06



Last update: **2019/05/14 16:00**

5.10.7) Arbeiten auf der Konsole

Gnome Terminal findet man unter *Weiter Anwendungen* .



prompt `user@linux-aosk:~` → hat folgende Bedeutung:

- `user` : Benutzername
- `linux-aosk` : Rechnernamen
- `~` : Homeverzeichnis

Kommandos zur Bearbeitung von Dateien

Obwohl unter KDE und Gnome moderne Dateimanager zur Verfügung stehen, verwenden erfahrene Linux-Anwender oft noch immer diverse, text-orientierte Kommandos.

Kommando	Beschreibung	Kommando in DOS
Hilfe		
<code>man Befehl</code>	Hilfe zum Kommando	
<code>Befehl - - help</code>	Hilfe zum Kommando	
Als Root		
<code>su</code>	wechselt als Root (Passwort eingeben)	
<code>sudo</code>	einen Befehl als Root ausführen	
Verzeichnisbaum		
<code>cd</code>	wechselt das aktuelle Verzeichnis	
<code>cd /</code>	wechselt ins root-Verzeichnis	

Kommando	Beschreibung	Kommando in DOS
Hilfe		
ls	zeigt alle Dateien des aktuellen Verzeichnisses an	dir
ls -l	zeigt eine detaillierte Liste	
ls -a	zeigt versteckte Dateien an	
mkdir	erzeugt ein neues Verzeichnis	md
rmdir	löscht Verzeichnisse	rd
pwd	zeigt aktuellen Pfad an	
Joker		
*	steht für eine beliebige Anzahl von beliebigen Zeichen	
?	steht für ein beliebiges Zeichen	
Dateien		
mv quelle ziel	verschiebt Dateien bzw. ändert ihren Namen	move
cp quelle ziel	kopiert Dateien	copy
cp ordner ziel -r	kopiert gesamten Ordner inkl. aller Unterordner an Ziel	
cat	zeigt Dateiinhalt an	type
less	öffnet Anzeigeprogramm	
more	zeigt Dateiinhalt seitenweise an	
touch Dateiname	erstellt leere Datei	
mcedit Dateiname	öffnet Datei in einem Editor zur Bearbeitung	edit
vim Dateiname	öffnet Datei mit dem Editor VIM zur Bearbeitung	
vimtutor	Tutorial zum Erlernen vom Editor VIM	
rm	löscht Dateien	del
rm unterordner -r	löscht gesamten Unterordner inkl. aller Dateien	
find -name dateinamen	sucht Dateien nach Namen	
Packen und Komprimieren von Verzeichnissen und Dateien		
tar	vereint mehrere Dateien (und Verzeichnisse) in einer Datei	
tar -t	Inhalt eines Archivs anzeigen	
tar -x	Dateien aus Archiv holen	
tar -c	neues Archiv erzeugen	
tar -f	um Namen des Archiv anzugeben	
tar -xvjf	entzippen	

Weitere Befehle

- <http://www.admintalk.de/konsolenbefehle.php>
- <http://www.shellbefehle.de/befehle/>

Übung 1

1. Erstelle in deinem Home-Directory einen Ordner uebungen.
2. Speichere das File [uebung1.tar](#) in diesen Ordner!
3. Entpacke das Archiv mit Hilfe von `tar -xvjf uebung1.tar`. Welche Verzeichnisse und Dateien befinden sich nun in deinem Home-Directory?
4. Gib den Befehl `./hallo` ein.

5. Finde die Datei `ichbinhier`.
6. Wechsle in das Verzeichnis, in dem sich die Datei befindet.
7. Erstelle ein Verzeichnis mit dem Namen `backup` in deinem Homedirectory.
8. Kopiere das gesamte Verzeichnis `uebung1` in das Verzeichnis `backup`.
9. Lösche das Verzeichnis `uebung1`.
10. Erstelle ein Verzeichnis mit dem Namen `aufgabe` und wechsle hinein.
11. Erstelle drei leere Dateien `datei1` bis `datei3`.
12. Öffne mit einem Editor `datei1` und gib drei Zeilen Text ein. Speicher ab!
13. Lasse dir die Datei mit einem entsprechendem Kommando ausgeben!
14. Gib den Befehl `cat datei1` ein. Was passiert?
15. Wechsle in die grafische Oberfläche!
16. Orientiere dich an der Oberfläche!
17. Versuche den Bildschirmhintergrund umzustellen.
18. Öffne ein Konsolenfenster. Lösche darin den gesamten Ordner `uebungen` inklusive Unterverzeichnis.

Übung 2

Grundkurs 1-3 & 7, Für Experten 4-6

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_07



Last update: **2019/05/14 16:00**

5.10.8) Shell Scripts sh (Bourne Shell)

Shell-Scripts (Kommandoprozeduren) sind unter Unix das Analogon zu Batch-Dateien (Stapeldateien) von MS-DOS, sie sind jedoch wesentlich leistungsfähiger. Ihre Syntax hängt allerdings von der verwendeten „Shell“ ab. In diesem Artikel werden Bourne Shell (sh) Scripts betrachtet. Sie gelten im allgemeinen als zuverlässiger als csh-Scripts.

Ein Shell-Script ist eine Textdatei, in der Kommandos gespeichert sind. Es stellt selbst ein Kommando dar und kann wie ein Systemkommando auch mit Parametern aufgerufen werden. Shell-Scripts dienen der Arbeitserleichterung und (nach ausreichenden Tests) der Erhöhung der Zuverlässigkeit, da sie gestatten, häufig gebrauchte Sequenzen von Kommandos zusammenzufassen.

Aufruf

Bourne Shell Scripts lassen sich generell wie folgt aufrufen:

```
sh myscript
```

Im allgemeinen ist es aber praktischer, die Datei als ausführbar anzumelden (execute permission), mittels

```
chmod +x myscript
```

Das Shell-Script läßt sich dann wie ein „normales“ (binäres) Kommando aufrufen:

```
myscript
```

Vorausgesetzt ist hierbei allerdings, daß das Betriebssystem das Script mit der richtigen Shell abarbeitet. Moderne Unix-Systeme prüfen zu diesem Zweck die erste Zeile. Für die sh sollte sie folgenden Inhalt haben:

```
#!/bin/sh
```

Obwohl das Doppelkreuz normalerweise einen Kommentar einleitet, der vom System nicht weiter beachtet wird, erkennt es hier, daß die „sh“ (mit absoluter Pfadangabe: /bin/sh) eingesetzt werden soll.

Ein Beispiel

```
#!/bin/sh
# Einfaches Beispiel
echo Hallo, Welt!
echo Datum, Uhrzeit und Arbeitsverzeichnis:
date
pwd
```

```
echo Uebergabe-Parameter: $*
```

Das vorstehende einfache Script enthält im wesentlichen normale Unix-Kommandos. Abgesehen von der ersten Zeile liegt die einzige Besonderheit im Platzhalter „\$*“, der für alle Kommandozeilen-Parameter steht.

Testen eines Shell-Scripts

```
sh -n myscript
```

Syntax-Test (die Kommandos werden gelesen und geprüft, aber nicht ausgeführt)

```
sh -v myscript
```

Ausgabe der Shell-Kommandos in der gelesenen Form

```
sh -x myscript
```

Ausgabe der Shell-Kommandos nach Durchführung aller Ersetzungen, also in der Form, wie sie ausgeführt werden

Kommandozeilen-Parameter

```
$0
```

Name der Kommandoprozedur, die gerade ausgeführt wird

```
$#
```

Anzahl der Parameter

```
$1
```

erster Parameter

```
$2
```

zweiter Parameter

```
$3
```

dritter ... Parameter

```
$*
```

steht für alle Kommandozeilen-Parameter (\$1 \$2 \$3 ...)

\$@

wie \$* (\$1 \$2 \$3 ...)

\$\$

Prozeßnummer der Shell (nützlich, um eindeutige Namen für temporäre Dateien zu vergeben)

\$-

steht für die aktuellen Shell-Optionen

\$?

gibt den Return-Code des zuletzt ausgeführten Kommandos an (0 bei erfolgreicher Ausführung)

\$!

Prozessnummer des zuletzt ausgeführten Hintergrund-Prozesses

Beispiel

```
#!/bin/sh
# Variablen
echo Uebergabeparameter: $*
echo user ist: $USER
echo shell ist: $SHELL
echo Parameter 1 ist: $1
echo Prozedurname ist: $0
echo Prozessnummer ist: $$
echo Anzahl der Parameter ist: $#
a=17.89          # ohne Luecken am = Zeichen
echo a ist $a
```

Prozesssteuerung

Bedingte Ausführung: if

```
if [ bedingung ]
then kommandos1
else kommandos2
fi
```

Anmerkungen: „fi“ ist ein rückwärts geschriebenes „if“, es bedeutet „end if“ (diese Schreibweise ist eine besondere Eigenheit der Bourne Shell). Die „bedingung“ entspricht der Syntax von test, siehe auch weiter unten. Im if-Konstrukt kann der „else“-Zweig entfallen, andererseits ist eine Erweiterung

durch einen oder mehrere „else if“-Zweige möglich, die hier „elif“ heißen:

```
if [ bedingung1 ]
    then kommandos1
elif [ bedingung2 ]
    then kommandos2
else kommandos3
fi
```

Die Formulierung

```
if [ bedingung ]
```

ist äquivalent zu

```
if test bedingung
```

Alternativ ist es möglich, den Erfolg eines Kommandos zu prüfen:

```
if kommando
```

(beispielsweise liefert das Kommando „true“ stets „wahr“, das Kommando „false“ hingegen „unwahr“)

Wichtige Vergleichsoperationen (test)

Hinweis: Es ist unbedingt notwendig, daß alle Operatoren von Leerzeichen umgeben sind, sonst werden sie von der Shell nicht erkannt! (Das gilt auch für die Klammern.)

Zeichenketten

```
"s1" = "s2"
```

wahr, wenn die Zeichenketten gleich sind

```
"s1" != "s2"
```

wahr, wenn die Zeichenketten ungleich sind

```
-z "s1"
```

wahr, wenn die Zeichenkette leer ist (Länge gleich Null)

```
-n "s1"
```

wahr, wenn die Zeichenkette nicht leer ist (Länge größer als Null)

(Ganze) Zahlen

$n1 -eq n2$

wahr, wenn die Zahlen gleich sind

$n1 -ne n2$

wahr, wenn die Zahlen ungleich sind

$n1 -gt n2$

wahr, wenn die Zahl $n1$ größer ist als $n2$

$n1 -ge n2$

wahr, wenn die Zahl $n1$ größer oder gleich $n2$ ist

$n1 -lt n2$

wahr, wenn die Zahl $n1$ kleiner ist als $n2$

$n1 -le n2$

wahr, wenn die Zahl $n1$ kleiner oder gleich $n2$ ist

Sonstiges

!

Negation

-a

logisches „und“

-o

logisches „oder“ (nichtexklusiv; -a hat eine höhere Priorität)

\(... \)

Runde Klammern dienen zur Gruppierung. Man beachte, daß sie durch einen vorangestellten Backslash, \, geschützt werden müssen.

-f filename

wahr, wenn die Datei existiert. (Weitere Optionen findet man in der man page zu test)

Beispiel:

```
#!/bin/sh
# Interaktive Eingabe, if-Abfrage
echo Hallo, user, alles in Ordnung?
echo Ihre Antwort, n/j:
read answer
echo Ihre Antwort war: $answer
# if [ "$answer" = "j" ]
if [ "$answer" != "n" ]
    then echo ja
    else echo nein
fi
```

Mehrfachentscheidung: case

```
case var in
    muster1) kommandos1 ;;
    muster2) kommandos2 ;;
    *) default-kommandos ;;
esac
```

Anmerkungen: „esac“ ist ein rückwärts geschriebenes „case“, es bedeutet „end case“. Die einzelnen Fälle werden durch die Angabe eines Musters festgelegt, „muster)“, und durch ein doppeltes Semikolon abgeschlossen. (Ein einfaches Semikolon dient als Trennzeichen für Kommandos, die auf derselben Zeile stehen.) Das Muster „*)“ wirkt als „default“, es deckt alle verbleibenden Fälle ab; die Verwendung des default-Zweiges ist optional.

Beispiel:

```
#!/bin/sh
# Interaktive Eingabe, Mehrfachentscheidung (case)
echo Alles in Ordnung?
echo Ihre Antwort:
read answer
echo Ihre Antwort war: $answer
case $answer in
    j*|J*|y*|Y*) echo jawohl ;;
    n*|N*) echo nein, ueberhaupt nicht! ;;
    *) echo das war wohl nichts ;;
esac
```

Schleife: for

```
for i in par1 par2 par3 ...
do kommandos
done
```

Anmerkungen: Die for-Schleife in der Bourne Shell unterscheidet sich von der for-Schleife in üblichen Programmiersprachen dadurch, daß nicht automatisch eine Laufzahl erzeugt wird. Der Schleifenvariablen werden sukzessive die Parameter zugewiesen, die hinter „in“ stehen. (Die Angabe „for i in \$*” kann durch „for i“ abgekürzt werden.)

Beispiel:

```
#!/bin/sh
# Schleifen: for
echo Uebergabeparameter: $*
# for i
for i in $*
do echo Hier steht: $i
done
```

Schleife: while und until

```
while [ bedingung ]
do kommandos
done
until [ bedingung ]
do kommandos
done
```

Anmerkung: Bei „while“ erfolgt die Prüfung der Bedingung vor der Abarbeitung der Schleife, bei „until“ erst danach. (Anstelle von „[bedingung]“ oder „test bedingung“ kann allgemein ein Kommando stehen, dessen Return-Code geprüft wird; vgl. die Ausführungen zu „if“.)

Beispiel:

```
#!/bin/sh
# Schleifen: while
# mit Erzeugung einer Laufzahl
i=1
while [ $i -le 5 ]
do
echo $i
i=`expr $i + 1`
done
```

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:05:05_10:05_10_08



Last update: **2019/05/14 16:00**

6) Hardware

Hardware ist der Oberbegriff für die **physischen Komponenten (die elektronischen und mechanischen Bestandteile)** eines datenverarbeitenden Systems, als **Gegenteil zu Software (den Programmen und Daten)**

Wortherkunft

„Hardware“ kommt ursprünglich aus dem Englischen und bedeutet übersetzt Eisenwaren.

Hardware vs. Software

- Hardware = sind alle greifbaren/sichtbaren Elemente eines PCs
- Software = Programme und Daten, also nicht greifbare Elemente eines PCs

Überblick

- [4.1\) EVA-Prinzip](#)
- [4.2\) Von-Neumann-Architektur](#)
- [4.3\) HW-Komponenten](#)
- [4.4\) Schnittstellen](#)
- [4.5\) Fragen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:06

Last update: **2019/05/14 16:04**



Textverarbeitung & Zeichencodierung

Theorie

Textverarbeitung ist die Erstellung und Bearbeitung von schriftlichen Texten mit Hilfe von Programmen/Applikationen.

Bis weit ins 19. Jahrhundert war es üblich, Texte (sofern sie nicht gedruckt wurden) mit der Hand zu schreiben. Nachdem im 19. Jahrhundert die ersten praktisch verwendbaren Schreibmaschinen auf den Markt kamen, begann sich dies jedoch allmählich zu ändern. Nach und nach setzte sie sich gegenüber der Handschrift durch, ihre **Vorteile gegenüber der Handschrift**:

- **Klareres Schriftbild**
- **Normierte Zeichengrößen**
- **Schnelleres Schreiben**

Vor allem im Bürobereich fanden Schreibmaschinen deshalb große Verbreitung. Dies hatte **enormen Einfluss auf die Arbeitswelt**: Die bis dahin oft gebräuchlichen Stehpulte wurden durch Schreibtische abgelöst, **das Schreiben von einer stehenden zu einer sitzenden Tätigkeit**. Dieser Prozess lief parallel zu anderen Entwicklungen in der Mechanisierung der Büroarbeit (Einführung von Rechenmaschinen, Hollerithmaschine und Telefon).

Mit der zunehmenden Verbreitung von PCs in den 1990er Jahren verschwanden Schreibmaschinen jedoch vom Markt.

PCs waren **universell einsetzbar** und **nicht nur auf Textverarbeitung** beschränkt, bei zunehmend günstigerem Preis-Leistungs-Verhältnis zugunsten des PCs. Die Fähigkeiten von Textverarbeitungen erforderten zunehmend vollwertige Computer, vor allem nach dem Aufkommen grafischer Benutzeroberflächen. Seit **Einführung des Personal Computers** hat sich der **Bereich der Textverarbeitung rasant entwickelt**. Das heutige Ergebnis sind Programme, bei denen der reine Textverarbeitungsteil programmiertechnisch wohl noch den geringsten Aufwand erforderte. Vielmehr wurden sie angereichert mit Funktionen, die zuvor einzeln von anderen Programmen und oft auch anderen Herstellern bezogen werden mussten. **Heutige Textverarbeitungsprogramme** integrieren die reine Texterfassung mit den **Möglichkeiten der Grafikeinbindung, Tabellenerstellung, Formelgenerierung oder der Präsentationsgrafik**.

Die Einführung der grafischen Benutzeroberflächen wie die des Apple Macintosh 1984 und später Windows ermöglichte erst den Schritt in diese Richtung, verhalf sie doch den Programmen, auf Techniken wie dem **Datenaustausch über eine zentrale Zwischenablage**, dem dynamischen Datenaustausch oder dem Objekt-verbinden-und-einbetten zurückzugreifen. Die Möglichkeit, den Text gemäß dem Prinzip **WYSIWYG (engl. What You See Is What You Get)** bereits auf dem Bildschirm so präsentiert zu bekommen, wie er auch später ausgedruckt erscheint (Druckdarstellung), war dabei von großer Bedeutung.

WYSIWYG

WYSIWYG ist das Akronym für den Grundgedanken **What You See Is What You Get** – auch als **Echtzeitdarstellung** bekannt (beziehungsweise, bezogen auf die Bildschirmdarstellung,

Echtbilddarstellung). Bei echtem WYSIWYG wird ein Dokument während der Bearbeitung am Bildschirm genauso angezeigt, wie es bei der Ausgabe über ein anderes Gerät, z. B. einen Drucker, aussieht. Der Begriff wurde zuerst Anfang der 1980er im Zusammenhang mit Computer-Drucksatz-Systemen (Desktoppublishing) und Textverarbeitungsprogrammen verwendet.

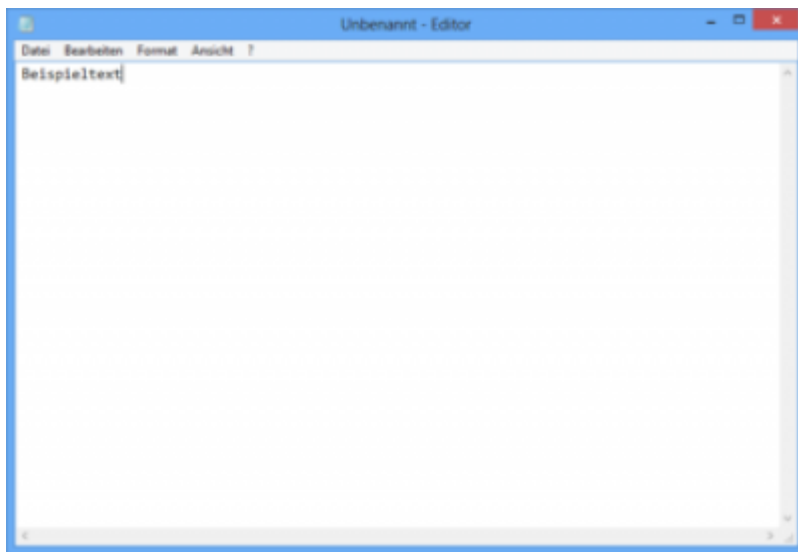
Texteditoren und ASCII-Editoren

Die ersten **Textverarbeitungsprogramme**, treffender bezeichnet als „Texterfassungsprogramme“, waren **einfache Texteditoren**, die sich aus dem **Zeichenvorrat nichtgenormter Zeichensätze** bedienten. Somit waren nicht nur die **Editoren an den zugehörigen Computer (bzw. das zugehörige Betriebssystem) gebunden**, sondern auch die darauf erstellten Texte, die bei Bedarf erst für andere Computersysteme in andere Zeichensätze konvertiert werden mussten.

Erst mit der Einführung **genormter Zeichensätze (z.B. ASCII)** kann man von **ASCII-Editoren** sprechen, die sich aus dem Zeichenvorrat der **standardisierten ASCII-Tabelle** bedienen. Damit wurden auch die Editorprogramme selbst grundlegend **kompatibel zu gleichartiger Hardware unter den meisten gängigen Betriebssystemen**.

Der Befehlsumfang einfacher ASCII-Editoren ging in der Regel über das Speichern, Laden und Drucken nicht hinaus.

Ein Beispiel für einen reinen ASCII-Editor ist der Editor unter Microsoft Windows.



Erweiterte Programme bieten jedoch neben Funktionen wie Textmarkierung oder das Ausschneiden, Kopieren und Versetzen von Textabschnitten an: **Syntaxhervorhebung und Makroprogrammierung**. Besonders **komplexe Editoren** wie **Emacs und Vim** unter Linux kommen einer modernen Textverarbeitung unter den Einschränkungen eines textorientierten Betriebssystems bereits sehr nahe und berücksichtigen die Bedürfnisse von Systemadministratoren oder Programmierern, etwa durch **Verknüpfung mit Compilern** und anderer **externer Software**.

⇒ Die bezeichnet man als „**ZEICHENCODIERUNG**“.

Dazu wird jedem Schriftzeichen eine Zahlenreihe aus Nullen und Einsen zugeordnet.

Beispiel:

- die Zahlenreihe 1000001 ⇒ hat den Dezimalwert 65 und steht für den großen Buchstaben „A“
- die Zahlenreihe 1100001 ⇒ hat den Dezimalwert 97 und steht für den kleinen Buchstaben „a“
- die Zahlenreihe 1000010 ⇒ hat den Dezimalwert 66 und steht für den großen Buchstaben „B“
- die Zahlenreihe 1100010 ⇒ hat den Dezimalwert 98 und steht für den kleinen Buchstaben „b“, usw.

ASCII

Grundsätzlich ist die Zuordnung der Zahl zum Buchstaben vollkommen willkürlich wählbar, und daher entwickelten sich in der **Anfangszeit der Computer in Amerika mehrere unterschiedliche Zuordnungsvorschriften**. Von diesen haben sich zwei heute gehalten:

- der American Standard Code for Information Interchange (ASCII)
- Extended Binary Coded Decimals Interchange Code (EBCDIC)

ASCII-TABELLE

Dezimal	Hexadezimal	Binär	Oktal	ASCII-Character
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	\$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135]
94	5E	1011110	136	^
95	5F	1011111	137	_
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

Drückt man also eine Taste der Tastatur, erkennt der Computer, welche Taste gedrückt wurde, und ordnet ihr aufgrund des konfigurierten Tastaturlayouts den Bytewert des Buchstabens zu, der sich dahinter verbirgt. Bei der Ausgabe des Buchstabens auf dem Bildschirm wird der Bytewert umgewandelt in die Anweisung, wie der entsprechenden Buchstabe zu malen ist, und er erscheint dann auf dem Bildschirm. Wie der Buchstabe konkret aussieht, entscheidet dabei die **Malanweisung der Schriftart**. **Ändert man die Schriftart**, wird eine **neue Malanweisung** verwendet, aber die **Bytewerte der Zeichen bleiben identisch**.

Diese betagten Codes wie ASCII und EBCDIC haben jedoch eine Schwäche: Mit ihnen lassen sich, da sie einem Buchstaben ein Byte zuordnen, **maximal 256 verschiedene Zeichen** darstellen. Dieser Zeichenvorrat **reicht gerade für das lateinische Alphabet** und **einige zusätzliche Zeichen**. Als die Codes in den **1960ern** entstanden, war das für die englisch sprechenden Amerikaner **ausreichend**. Als ASCII entstand, kam man – zumindest in der Datenverarbeitung im englischen Sprachraum – mit 128 verschiedenen Zeichen aus, es reichten daher **7 Bit für die Codierung des Zeichensatzes**. Das **8. Bit** eines Bytes wurde **nicht benutzt** oder für verschiedene andere Zwecke benutzt!

Die erste, nicht hundertprozentig befriedigende Lösung für dieses Problem bestand darin, für andere Alphabete neue Zuordnungen zu erstellen. Dies bedeutete bei der interkulturellen Kommunikation aber einen kleinen Mehraufwand. Wollte beispielsweise ein Russe einem Amerikaner einen russischen Text zukommen lassen, so reichte es nicht, einfach die Textdatei weiterzugeben. Der Absender musste zusätzlich die für das kyrillische Alphabet verwendete Codierung angeben. Mit der amerikanischen Codierung würde der Computer des Empfängers die Bytes als lateinische Buchstaben interpretieren und somit einen sinnlosen Buchstabensalat produzieren.

ISO-8859-Familie

Im **westeuropäischen Sprachraum** verbreiteten sich insbesondere **zwei Codierungen aus der sogenannten ISO-8859-Familie**:

- ISO-8859-1 und
- ISO-8859-15.

Den Codierungen der ISO-8859-Familie ist gemeinsam, dass sie nur 256 verschiedene Zeichen codieren können, da ein Zeichen immer mit einem Byte codiert wird. Jede dieser Codetabellen versucht, möglichst alle Zeichen möglichst vieler Schriften zu speichern. Sie **verwenden alle die ASCII-Codetabelle für die Zeichen 0 bis 127** und definieren **zusätzlich die Werte 160 bis 255**. Sofern in diesem Bereich das gleiche Zeichen in verschiedenen Schriften vorkommt, überlappen sie sich zumeist.

Upper part of ISO 8859-1 character set

160 NBSP	161 ı	162 Œ	163 £	164 ¤	165 ¥	166 ¦	167 §	168 ¨	169 ©	170 ª	171 «
172 ¬	173 SHY	174 ®	175 ¯	176 °	177 ±	178 ²	179 ³	180 ´	181 µ	182 ¶	183 ·
184 ¸	185 ¹	186 º	187 »	188 ¼	189 ½	190 ¾	191 ¿	192 À	193 Á	194 Â	195 Ã
196 Ä	197 Å	198 Æ	199 Ç	200 È	201 É	202 Ê	203 Ë	204 Ì	205 Í	206 Î	207 Ï
208 Ð	209 Ñ	210 Ò	211 Ó	212 Ô	213 Õ	214 Ö	215 ×	216 Ø	217 Ù	218 Ú	219 Û
220 Ü	221 Ý	222 Þ	223 ß	224 à	225 á	226 â	227 ã	228 ä	229 å	230 æ	231 ç
232 è	233 é	234 ê	235 ë	236 ì	237 í	238 î	239 ï	240 ð	241 ñ	242 ò	243 ó
244 ô	245 õ	246 ö	247 ÷	248 ø	249 ù	250 ú	251 û	252 ü	253 ý	254 þ	255 ÿ

Der universale Code: Unicode

Mit **Unicode** werden so ziemlich **alle Zeichencodierungsprobleme** dieser Welt **gelöst**. Und weil Webseiten potentiell mit allen Schriften dieser Welt genutzt werden, ist für HTML 4.0 und XML (und damit auch XHTML) festgelegt, dass grundsätzlich alle in Unicode definierten Zeichen vorkommen dürfen.

Mit dieser Festlegung kann man also jedes der mittlerweile über **100.000 Unicode-Zeichen** in seinem Text verwenden. Aber Unicode regelt nicht nur die Codierung aller Schriftzeichen dieser Welt, sondern kennt zu jedem Zeichen dutzende definierte Eigenschaften. Ein Zeichen besitzt mindestens folgende Informationen:

- der eindeutige Unicode-Name des Zeichens
- den zugeordneten Codepoint (die „Nummer“ des Zeichens)
- eine generelle Kategorisierung (Buchstabe, Zahl, Symbol, Interpunktion, ...)
- weitere grundsätzliche Charakteristiken (Whitespace, Bindestrich, alphabetisch, ideographisch, Nicht-Zeichen, veraltet, ...)
- anzeigerelevante Informationen (Schreibrichtung, Form, Spiegelung, Breite)
- Groß-/Kleinschreibung/Zahlenwert und -typ bei Zahlzeichen
- Trennfunktionen (zwischen Wörtern, Zeilen, Sätzen,...)
- usw.

Die meisten Eigenschaften eines Zeichens sind nur für die interne Behandlung durch den Computer relevant. Beispielsweise werden arabische Schriftzeichen automatisch korrekt von rechts nach links geschrieben, oder der Zeilenumbruch nutzt die Stellen, an denen eine Wortgrenze z. B. durch ein normales Leerzeichen existiert. Dieser kleine Exkurs soll aber verdeutlichen, dass die Bytewerte der Buchstaben für den Computer viel mehr bedeuten können, als nur den Buchstaben an sich.

Der oben erwähnte Codepoint eines Unicode-Zeichens ist nur **eine abstrakte Nummer**. Die Schreibweise dieser Nummer im Unicode-Standard erfolgt in **hexadezimalen Zahlen mit vorangestelltem „U+“**. Der Codepoint legt noch keinerlei computerkompatible Darstellung fest, dies ist Aufgabe des Codierschemas. Da die Unicode-Codepoints von U+0000 bis U+10FFFF (hexadezimale Zahlendarstellung), mit einer beabsichtigten Lücke zwischen U+D7FF und U+E000, reichen, sind für eine vollständige Codierung des gesamten Codepoint-Bereichs als Binärzahl mindestens 3 Byte erforderlich.

Folgende Speichervarianten von Unicode sind heute gängig:

- **UTF-8** (8 Bits \Rightarrow 1 Byte)
- **UTF-16** (16 Bits \Rightarrow 2 Byte)
- **UTF-32** (32 Bits \Rightarrow 3 Byte)

Praxistipp

Empfehlung: Verwenden Sie, wo immer es geht, UTF-8! Es hat viele Vorteile, bei der Wahl der codierbaren Zeichen nicht eingeschränkt zu sein. Selbst wenn ein Webauftritt garantiert nur für eine einzige Sprache (z. B. Deutsch) gestaltet werden muss, bietet nur Unicode die komplette Freiheit, wirklich alle Zeichen – insbesondere Interpunktionszeichen wie „typografische Anführungszeichen“ oder eben das Eurozeichen – beliebig verwenden zu können. Eingestreute Fremdwörter in einer anderen Sprache sind genauso problemlos möglich wie die Verwendung eher exotischerer Zeichen wie den Brüchen $\frac{1}{3}$, $\frac{2}{3}$, $\frac{1}{8}$, $\frac{3}{8}$, $\frac{5}{8}$, $\frac{7}{8}$.

Viele Programme können schon seit langer Zeit mit UTF-8 umgehen: Server, Browser, Datenbanken, Programmiersprachen, Editoren. Der einzige Grund, auf UTF-8 zu verzichten, ist die Existenz von alten Systemen oder Daten, die nicht in UTF-8 gespeichert sind und bei denen es zu aufwendig wäre, sie nach UTF-8 zu konvertieren.

Begrifflichkeiten

Für die Zeichen-Problematik gibt es eine Reihe Begrifflichkeiten, die allerdings oftmals falsch angewendet werden. Nachfolgend der Versuch einer Aufklärung:

Zeichensatz

Ein **Zeichensatz (character set, kurz charset)** ist die **Gesamtheit der zur Verfügung stehenden Zeichen**. Ein Zeichensatz ist ein eher abstraktes Gebilde, das nur die Zeichen selbst und eine Reihenfolge beschreibt, **nicht jedoch eine konkrete Abbildung auf Byte-Werte**. Das ist Aufgabe der **Zeichencodierung**.

Beispielsweise ist **Unicode ein Zeichensatz**, **UTF-8** hingegen ist eine **Zeichencodierung**.

Codepoint

Zeichen in einem Zeichensatz (z. B. ASCII oder Unicode) werden in einer **bestimmten Reihenfolge in einem Coderaum (code space)** angeordnet. Die Position eines Zeichens ist der Codepoint (zu Deutsch etwa „**Codeposition**“).

Das heißt die Codeposition ist der Zahlenwert eines Zeichens im Coderaum. \

Bsp.:

A hat die Codeposition 65 im Coderaum ASCII.

Ein **Zeichensatz mit Codepoints** ist ein **codierter Zeichensatz (coded character set)**.

Schriftart

Für die grafische Darstellung von Zeichen wird eine Schriftart verwendet, die in der Regel eine in sich **konsistente Gestaltung** aufweist, beispielsweise für die **Strich-Dicke** oder **Verzierungen**. Arial, Times New Roman oder Courier sind Beispiele für Schriftarten.

Schriftart ist im engeren Sinne nach eher ein **Synonym zu Schriftschnitt**, also bspw. **Helvetica 24pt fett**. Da Schriftarten wie Arial, Helvetica, Times Roman usw. oft aus mehreren Schriftschnitten für Breiten (schmal, breit, ...), Strichstärken (leicht, normal, fett, ...) und Zeichenlagen (normal, kursiv, ...) bestehen, verwendet man auch den Begriff **Schriftfamilie**, der nicht genau von **Schriftart** **abzugrenzen** ist.

Vanessa - Ihr Wunschtext - 0123456789

SL - Ihr Wunschtext - 0123456789

Arndt - Ihr Wunschtext - 0123456789

Champagne - Ihr Wunschtext - 0123456789

Edwardian - Ihr Wunschtext - 0123456789

Nuance - Ihr Wunschtext - 0123456789

Comic - Ihr Wunschtext - 0123456789

Harrington - Ihr Wunschtext - 0123456789

Mono - Ihr Wunschtext - 0123456789

Trianti - Ihr Wunschtext - 0123456789

Times - Ihr Wunschtext - 0123456789

Englisch - Ihr Wunschtext - 0123456789

Times - Ihr Wunschtext - 0123456789

Kriterien für Schriftarten

Die Kriterien für die Verwendung einer bestimmten Schriftart können sein:

- **Verfügbarkeit:** Die Windows-Schriftarten wie Arial zum Beispiel sind gerade deshalb weit verbreitet, weil sie auf jedem Windows-System verfügbar sind und entsprechend formatierte Dokumente auch unter mehreren Benutzern leicht ausgetauscht werden können
- **Emotionale Wirkung:** Ein Text kann die unterschiedlichsten Zwecke erfüllen. Entsprechend seiner Aufgabe wird er in einer dazu passenden Schrift gesetzt. Im Verkehrsbereich finden vorwiegend Grotesken, wie die Univers, Anwendung, in belletristischer Literatur wird gerne auf Renaissance-Antiqua wie die Garamond zurückgegriffen, und eine spielerischere Schriftart ist die Comic Sans MS, die auch als „Kinder-Schriftart“ benutzt wird.
- **Lesbarkeit:** Sie ist wichtig, um ein Buch entspannt lesen zu können oder weiter entfernte Straßenschilder zu entziffern. Das Auge des Lesers muss möglichst leicht die einzelnen Buchstaben eines Textes voneinander trennen und erkennen können. Einfluss auf die Lesbarkeit einer Schrift hat neben der Gestaltung der Wort- und Zeichenzwischenräume auch die Gestalt der Zeichen selbst.

Schriftgestalt

Serifen

Serifen sind **kleine Endstriche eines Buchstabens**, umgangssprachlich auch „**Füßchen**“ genannt. Sie bilden eine horizontale Linie, an der sich das **Auge des Lesers orientieren** kann. Daher eignen sich Serifenschriften, insbesondere die Antiqua, **besonders für gedruckten Fließtext (Bücher,**

Artikel). Bei Postern, Plakaten, Schildern usw. kommt es dagegen darauf an, auch auf größere Distanz einzelne Wörter zu entziffern. Hier werden wegen ihrer größeren Klarheit Schriften ohne Serifen eingesetzt.

Courier: Wi wi

Times: Wi wi

Arial: Wi wi

Verdana: Wi wi

Höhe

Neben der Tatsache, dass es zwei parallele Alphabete von Groß- und Kleinbuchstaben gibt, unterscheiden sich seine Buchstaben noch in weiteren Punkten. Zum Beispiel betrachtet man die Höhe der einzelnen Buchstaben. Zunächst einmal allen gemeinsam ist, dass sie eine Grundlinie teilen. Die **Grundlinie ist der Boden der Zeile** selbst, der bei Serifenschriften durch ihre Füße angedeutet wird. Auf dieser Grundlinie „stehen“ alle Buchstaben und gehen unterschiedlich weit über bzw. unter diese hinaus.



Laufweite

Die **Laufweite** einer Schrift beschreibt, wie groß die **horizontale Ausdehnung eines geschriebenen Textes** ist. Sie entsteht durch die Breite der einzelnen Buchstaben sowie den Abstand, den sie zueinander haben. Die Laufweite spielt **beim Schriftsetzen in Büchern, Zeitungen, Magazinen** usw. eine große Rolle, da sie bestimmt, wie „**ökonomisch**“ eine Schrift ist, das heißt, wie **platzsparend** sie ist. Um wertvollen Platz zu sparen, gibt es gerade für diesen Zweck entworfene Schmal-Schnitte einer Schrift (englisch „condensed“). Wesentlich seltener werden extra breite Schnitte verwendet (englisch „extended“), die meisten hiervon sind Grotesken (Sans Serifs). Der **Verwendungszweck von breiten Schnitten** ist weniger ökonomisch als **grafisch (für Überschriften, Plakate, Logos, Corporate Design usw.)**.

so close and yet so far away
so close and yet so far away
so close and yet so far away
so close and yet so far away

Proportionen

Im **Normalfall** sind die einzelnen **Zeichen einer Schriftart unterschiedlich breit**, ein „w“ nimmt also mehr Platz ein als ein „i“. Solche Schriftarten werden **proportional** genannt. Verbreitete Mitglieder dieser Gruppe sind zum Beispiel **Times**, **Arial** oder **Lucida**. Um die **Konstruktion von Schreibmaschinen und Computern zu vereinfachen**, kamen **nichtproportionale**, sogenannte **dicktengleiche Schriftarten zum Einsatz**, bei denen alle Zeichen eine identische Breite aufweisen. Die wohl bekanntesten dieser Schriften sind **Courier** und **Lucida Console**.



Glyphe

Der Begriff **Glyphe** wird gelegentlich für die **konkrete Darstellung eines Zeichens** verwendet. Die Schriftart bestimmt, welches Zeichen mit welcher Glyphe dargestellt wird.



Font

Ein **Font** ist die **Aufbereitung einer Schriftart für den Einsatz mit einem Computer**, also eine **Datei**, die eine **Schriftart beschreibt**.

Tastatur-Layout

Ein **Tastatur-Layout** oder eine **Tastaturbelegung** (auch Tastenlayout oder Tastenbelegung) **ordnet** einer Taste auf der **Tastatur einen Codepoint** zu.

Beispielsweise produziert die auf einer deutschen Tastatur mit „z“ beschriftete Taste mit einer deutschen Tastenbelegung den Unicode-Codepoint U+007A, also (erwartungsgemäß) ein „z“. Ändert man die Tastenbelegung auf US-amerikanisch, so erzeugt dieselbe Taste den Codepoint U+0079, also ein „y“. Mit einer russischen Belegung produzieren praktisch sämtliche Tasten andere Resultate, nämlich die Codepoints für kyrillische statt lateinischer Buchstaben, also z.B. „н“ (U+043D) statt „z“.

Textverarbeitungsprogramme

Textverarbeitungsprogramme bieten im Gegensatz zu reinen Texteditoren in der Regel **erweiterte Layout- und Formatierungsfunktionen** an. Neben der Textüberarbeitung erhöhen **Rechtschreibprüfung, Indexerstellung, Such- und Ersetzfunktionen** den Nutzen für den Anwender. **Formatvorlagen** vereinfachen zudem eine einheitliche Gestaltung der zu veröffentlichenden Dateien, Textbausteine die Einbindung von wiederkehrenden Inhalten.

Zeichenorientierte Textverarbeitung

Die Kategorie der im Funktionsumfang erweiterten zeichenorientierten Textprogramme (**Character Used Interface**) verwendet ebenfalls, wie die ASCII-Editoren auch, den normierten und **beschränkten ASCII-Zeichensatz** als Grundlage. Die fest definierten Zeichensätze werden sowohl für die Bildschirmdarstellung als auch für den Ausdruck verwendet. Der **Abstand der Zeichen untereinander ist fest vorgegeben**, wie auch die Zeichen selbst statisch sind. Diese „**Statik**“ bedeutet, dass alle **Zeichen des Zeichenvorrats vorgeformt** und **fertig zur Verfügung** stehen. Hier liegt der **große Vorteil der CUI-Programme** begründet: **die Arbeitsgeschwindigkeit**. Da alle **Zeichen in Größe und Form fertig** vorliegen, bedarf es **keiner ständigen**

Neuberechnung von deren Bildschirmdarstellung. Die **Hardware-Anforderungen** beim Einsatz von CUI-Programmen sind entsprechend **gering**. Der dadurch erkaufte Nachteil ist jedoch die heute gewünschte Darstellungsqualität, die fortwährende Ansicht als Seitenvorschau.

Grafisch orientierte Textverarbeitung

Die grafisch orientierten Programme basieren auf dem **GUI, dem Graphical User Interface**. Alle **Zeichen** sind in **Form und Größe variabel**. Am Bildschirm können also **Formatierungen und Schriftgrößen exakt so angezeigt werden**, wie sie formatiert wurden und später auch auf dem **Ausdruck** erscheinen. Ermöglicht wird dies durch die **getrennte Handhabung von Schriftzeichen für den späteren Ausdruck** zum einen und **Bildschirmschriften** zum anderen. Der Preis hierfür ist jedoch ein **enormer Rechenaufwand** für die ständige **Aktualisierung und Neuberechnung der Bildschirmanzeige**. **Wichtig** geworden sind deshalb neben der reinen **Prozessorleistung** des Rechners Komponenten wie die **Grafikkarte** und der Bildschirmspeicher. Das Druckergebnis soll also exakt dem entsprechen, was auf dem Bildschirm zu sehen ist (**WYSIWYG, What You See Is What You Get**). Oder anders ausgedrückt: **Alles, was und wie man es auf dem Bildschirm sieht, soll auch genauso ausgedruckt werden**.

Dateiformate

Jedes **Textverarbeitungsprogramm** hat bis jetzt noch **sein eigenes Dateiformat** für die Speicherung von Dokumenten. Eine gewisse **Standardisierung** ist durch das **Rich Text Format** erreicht worden, mit dem aber die **Einheitlichkeit des Layouts auf verschiedenen Rechnern nicht gewährleistet** ist.

Eines der am weitesten verbreiteten Dokumentenformate zum Austausch von Dokumenten, die nicht zur weiteren Bearbeitung vorgesehen sind, ist **PDF**. Dieses wird als **ergänzendes Standardformat** von immer mehr Textverarbeitungsprogrammen oder **als separater virtueller Druckertreiber als Exportfunktion** (d. h. zum Speichern) angeboten, kann unter den meisten gängigen Betriebssystemen (darunter auch einige ältere Versionen) eingelesen werden und **behält dabei das Aussehen**.

Nachteil: Man kann das **PDF-Dokument nicht mehr bearbeiten**.

Klassifikation von Textverarbeitungsprogrammen

Textverarbeitungsprogramme können nach unterschiedlichen Kriterien voneinander unterschieden und klassifiziert werden:

- Komplexität (vom einfachen Texteditor bis zur hochkomplexen Office-Software)
- Verwendungszweck (Textverarbeitungsprogramme für Briefe, Verfassen von Quellcode z.B. notepad++)
- Plattformunterstützung (Linux, Windows, iOS, Android, ...)
- Lizenzierung (Freeware oder Commercial)

Kritik

Das Verfassen am Bildschirm könne dazu führen, dass man einen **geringen Überblick** über den Text habe.

Texte würden **nicht mehr selbst erstellt und durchdrungen (und auch kaum noch gelesen)**, sondern mittels **Kopieren und Einfügen** zusammengesetzt. Dieses Phänomen zeige sich besonders bei **ergoogelten Plagiaten**

Praxis mit MS Word

- [Grundlagen](#)
- [Kopf und Fußzeile](#)
- [Formatvorlagen](#)
- [Tabulatoren](#)
- [Serienbrief](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07

Last update: **2019/05/10 14:32**



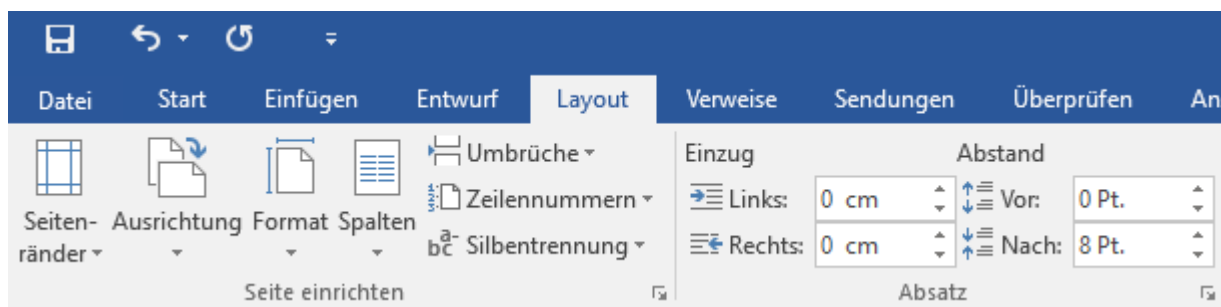
Textverarbeitung - Grundlagen

Layout

Seite einrichten

Mit dem Seitenformat definiert man eine Formatierung, die (meist) für das ganze Dokument gilt.

- Papierformat (im Menü Datei → Seite einrichten → Layout zu finden)
- Hoch- und Querformat (im Menü Datei → Seite einrichten → Seitenränder zu finden)
- Randeinstellungen (im Menü Datei → Seite einrichten findest du die wichtigsten Einstellungen)



Wichtige Tastenkombinationen

- **STRG + C** ... markierten Text kopieren
- **STRG + V** ... markierten Text einfügen
- **STRG + X** ... markierten Text ausschneiden
- **STRG + Mausrade** ... Einstellen der angezeigten Seitenfläche
- **Maus auf Blattkante → Symbol mit zwei Pfeilen** ... Leerfläche aus- bzw. einblenden
- **verkehrtes Pi-Symbol** ... Sichtbarmachen von Steuerzeichen (zB. Absatzende, Umbrüche,...)

Markierungen

Prinzipiell mit der Maus möglich, aber oftmals sind Shortcuts ganz nützlich!

- Markiere ein ganzes Dokument ⇒ **STRG+A**
- Markiere ein ganzes Wort ⇒ Doppelklick mit der Maus auf das Wort oder **STRG+SHIFT+Pfeiltaste rechts** bzw. **Pfeiltaste links**
- Markiere eine ganze Zeile ⇒ Mausklick links auf den Rand auf Höhe der gewünschten Zeile **STRG+SHIFT+ENDE**
- Markiere mehrere Zeilen untereinander ⇒ Mausklick links auf den Rand auf Höhe der gewünschten Zeile und nach unten oder oben ziehen **STRG+SHIFT+Pfeiltaste oben** oder **Pfeiltaste unten**

- Markiere mehrere einzelne Zeilen ⇒ Bei gedrückter **STRG + ein Mausklick links** auf die gewünschten Zeilen
- Markiere von der aktuellen Position alles bis zum Seitenanfang ⇒ **STRG+SHIFT+POS1**
- Markiere von der aktuellen Position alles bis zum Seitenende ⇒ **STRG+SHIFT+ENDE**

Umbrüche

Enter ... Absatzende

Strg + Enter ... Seitenumbruch

Shift + Enter ... Zeilenumbruch

Einfügen→Manueller Umbruch ... Menü zum Einfügen verschiedener Umbrüche

Zeichenformate

Um die Schrift verändern zu können gibt es drei Möglichkeiten:

- Auf der **Symbolleiste FORMAT** findest du die gängigsten Optionen
- Im Menü **FORMAT → Zeichen** findest du,
- genauso wie mit **RECHTER MAUSTASTE → Schriftart**, eine größere Auswahl um deinen Text zu formatieren

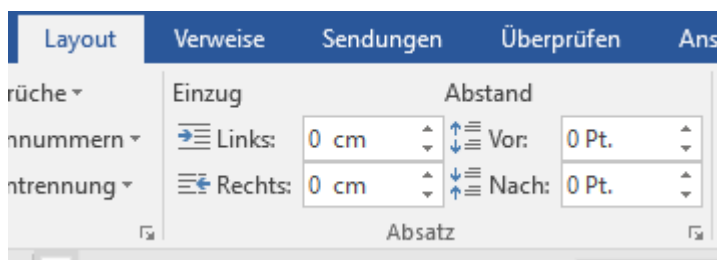
!! Der zuändernde Text muss dabei immer markiert sein !!

Textausrichtung und Absatzgestaltung

Die gewünschte Textausrichtung **links - rechts - zentriert - Blocksatz** kann mittels

- Symbolleiste FORMAT,
- im Menü FORMAT → Absatz oder
- mit RECHTER MAUSTASTE → Absatz eingestellt werden.

!! Der zuändernde Text muss dabei immer markiert sein !!



Aufzählung - Nummerierung

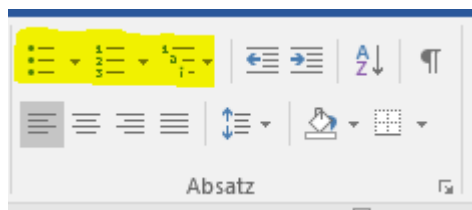
Ein Text kann durch Aufzählungszeichen und/oder Nummerierungen gegliedert werden.
Je nach Gliederungswunsch, kann:

- in der Symbolleiste FORMAT → Aufzählung / Nummerierung oder
- mit RECHTER MAUSTASTE → Aufzählung / Nummerierung eine Gliederung vorgenommen werden.

Dabei ist sowohl das Symbol bzw. die Nummerierung, sowie der Einzug frei wählbar.

Auch benutzerdefinierte Gliederung ist möglich!

!! Der zuändernde Text muss dabei immer markiert sein !!



From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07:07_01



Last update: **2019/05/10 14:32**

Kopf- und Fußzeile

Seitenzahl

Bei mehrseitigen Dokumenten möchte man manchmal unterschiedliche Kopf- und Fußzeilen für gerade und ungerade Seiten erstellen.

Unter Kopfzeile bearbeiten, bzw. Fußzeile bearbeiten gibt es die Möglichkeit, „Gerade und ungerade Seiten untersch.“ auszuwählen.

Weiters ist es möglich, die erste Kopf- bzw. Fußzeile anders zu gestalten.

Abschnittswechsel

Mithilfe von Abschnitten wird ein Dokument unterteilt. Die einzelnen Abschnitte eines Dokumentes nummeriert WORD fortlaufend. Abschnitte sind notwendig, wenn man innerhalb eines Dokumentes folgende Merkmale ändern will:

- Seitenränder, Seitenzahlen, Papierformat und Ausrichtung,
- Spaltenanzahl,
- Kopf- und Fußzeilen, sowie Fußnotennummerierung.

Mittels Seitenlayout - Umbrüche - Nächste Seite fügt man einen Abschnittsumbruch ein. Allerdings ist der neue Abschnitt mit dem vorigen verknüpft! Möchte man die Kopf- und Fußzeilen im neuen Abschnitt anders formatieren, so muss man im neuen Abschnitt die Verknüpfung aufheben unter: Kopf- oder Fußzeile bearbeiten - Mit voriger verknüpfen. Es ist darauf zu achten, dass Mit voriger verknüpfen **nicht** markiert ist!

Beispielsweise möchte man mit der Seitennummerierung erst ab Seite 4 beginnen. In diesem Fall muss zwischen Seite drei und Seite 4 ein Abschnittsumbruch eingefügt werden, sowie die Verknüpfung aufgehoben werden. Erst dann kann man die Kopf- und Fußzeilen der beiden Abschnitte unabhängig voneinander formatieren.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07:07_02

Last update: **2019/05/10 14:33**



Formatvorlagen

Anwenden von Formatvorlagen

Formatvorlagen verleihen Ihrem Dokument ein einheitliches, professionelles Aussehen.

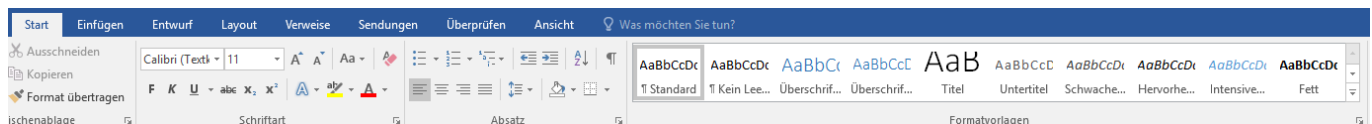
- **Markieren Sie den Text, den Sie formatieren möchten.**

Tipps: Wenn Sie den Cursor in einen Absatz setzen, wird die Formatvorlage auf den ganzen Absatz angewendet. Wenn Sie eine bestimmte Textpassage markieren, wird nur diese formatiert.

- **Zeigen Sie auf der Registerkarte Start auf eine Formatvorlage, um sich anzuschauen, wie der Text mit dieser Formatvorlage aussehen wird. Anwenden von Formatvorlagen-Optionen auf der Word-Multifunktionsleiste**

Tipps: Wenn die gewünschte Formatvorlage nicht angezeigt wird, klicken Sie auf die Schaltfläche Weitere Schaltfläche „Weitere“, um den Katalog zu erweitern.

- **Wählen Sie eine Formatvorlage aus.**



Nachdem Sie in ihrem Textdokument die Überschriften mit einer oder mehreren Formatvorlagen formatiert haben, können Sie daraus automatisch ein Inhaltsverzeichnis generieren.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07:07_03

Last update: 2019/05/10 14:35



Tabstopps

Mithilfe von Tabstopps können Sie einfach zu formatierende Dokumente erstellen. Mit den Optionen für das Dokumentlayout in Word können Sie beispielsweise ein Inhaltsverzeichnis oder einen Index erstellen, ohne einen einzigen Tabstopp setzen zu müssen.

Word stellt zudem vordefinierte Tabellen, Kopf- und Fußzeilen, Deckblätter und weitere Seitenlayoutoptionen bereit. Mit diesen Optionen erübrigt sich das Hinzufügen von Tabstopps.

Grundlegendes

Ganz gleich, welche Version von Word Sie verwenden, das Verfahren zum Setzen (Hinzufügen) von Tabstopps auf dem horizontalen Lineal ist immer dasselbe. Standardmäßig befinden sich keine Tabstopps auf dem Lineal, wenn Sie ein neues leeres Dokument öffnen. Es gibt jedoch eine standardmäßige Tabstoppweite, die auf jeweils 1,25 cm festgelegt ist.

Bevor Sie mit dem Setzen (Hinzufügen) von Tabstopps beginnen, möchten Sie sich möglicherweise mit der Tabstopppauswahl und den verfügbaren Tabstopps vertraut machen.



Die Tabstopppauswahl ist das kleine Feld am äußerst linken Rand des Lineals. Sie können auf die Tabstopppauswahl klicken, um jeden der im Folgenden beschriebenen Tabstopps auszuprobieren.

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07:07_04



Last update: **2019/05/10 14:35**

Serienbriefe

Ein Serienbrief ist ein Schreiben mit gleich bleibendem Text, welcher automatisch mit unterschiedlichen Anschriften bzw. Anreden versehen wird.

Elemente:

Datenquelle \longleftrightarrow Hauptdokument

Datenquelle: Sammlung von Adressen und personenbezogenen Angaben (Anrede, Geschlecht, Geburtsdatum etc.) (Excel, Word, Access-Datei)

Hauptdokument: allgemein gehaltener Text, der mit den Daten gefüllt wird.

1. Datenquelle erstellen (Excel-Tabelle, etc.)
2. Word: Sendungen – Seriendruck starten
 - Empfänger auswählen – Vorhandene Liste...
 - Brief schreiben – Seriendruckfelder nach Bedarf einfügen.
 - Regeln – Wenn-Dann-Sonst...
 - Alt+F9 → Formelansicht (erweitertes Bearbeiten)
 - Strg+F9 → geschwungene Klammern einfügen!

- [Serienbriefe Übungen 1](#)
- [Serienbriefe Übungen 2](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:07:07_05



Last update: **2019/05/10 14:39**

Tabellenkalkulation

Theorie

Eine Tabellenkalkulation ist eine Software für die interaktive Eingabe und Verarbeitung von numerischen und alphanumerischen Daten in Form einer Tabelle. Vielfach erlaubt sie zusätzlich die grafische Darstellung der Ergebnisse in verschiedenen Anzeigeformen.

Das Bildschirmfenster der Software ist dabei in Zeilen und Spalten eingeteilt. Je nach Programm bzw. Bedienungskonzept heißt dieser Bereich zum Beispiel Arbeitsblatt, Worksheet oder Spreadsheet. Jede Zelle der Tabelle kann eine Konstante (Zahl, Text, Datum, Uhrzeit ...) oder eine Formel enthalten. Für die Formeln stehen meist zahlreiche Bibliotheksfunktionen zur Verfügung. Die Formeln können Werte aus anderen Zellen benutzen.

Bei Änderung der referenzierten Zellen einer Formel aktualisiert die Software den erst angezeigten Wert der Formelzelle normalerweise automatisch, ggf. aber auch nur auf Anforderung. Werden Formeln eines Tabellenfeldes an andere Stellen kopiert, muss zwischen absolutem und relativem Zellbezug unterschieden werden. Formelzellen können auf andere Formelzellen verweisen. Mit diesem Prinzip können komplizierte Rechengänge mit vielen verknüpften Teil-Ergebnissen übersichtlich dargestellt werden.

Wozu?

Die Erfindung der elektronischen Tabellen zusammen mit Textverarbeitungs-Software und Datenbanken war zweifellos ein wesentlicher Faktor im Menschen den Wert von Mikrocomputern in den Anfangsjahren nach Apple und IBM begann Vermarktung Personalcomputer zu überzeugen. Seit dieser Zeit haben die stetig wachsende Vielseitigkeit und weiteren Anwendungen der Tabellenkalkulations-Software es in ein Produkt gemacht, die scheint fast unentbehrlich für Unternehmen und Privatanwender. Tabellen sind jetzt ein Standardbestandteil der Office-Suite-Pakete.

Funktionen

Kalkulationstabellen können mehr als einfache arithmetische Berechnungen durchführen. Eine Tabellenkalkulation übersetzen komplizierte Daten und Berichte in eine Kombination von Zahlen und Grafiken. Moderne Versionen enthalten eine umfangreiche Liste von finanzielle Rechner, z. B. Interesse Berechnungen, Kreditberechnung, auch Berechnungen für Schatzwechsel Preise. Statistische Funktionen von üblichen Berechnungen (Chi-Quadrat, Pearson-Koeffizient der Korrelation und Standardabweichung) auf abstruse Funktionen wie die hypergeometrische Verteilung und die Poisson-Verteilung Rückgabewerte benötigten mit keinen Schmerz. Es gibt mindestens 100 dieser Formeln in zeitgenössischen Tabellen enthalten.

Tabellen können als einfachen Datenbanken funktionieren. Durch das Einfügen von Daten und Anzahl in verschiedenen Spalten, können die Ergebnisse sortieren, gesucht oder gefiltert. Was-wäre-wenn-Analyse läuft, wenn Daten aus Zellen in einem Arbeitsblatt. Es gibt auch mathematische und trigonometrische Funktionen. Ein Benutzer kann eine vereinfachte Mailing-Liste in einer

Kalkulationstabelle erstellen, durch die Eingabe von Namen und Adressen in einzelnen Spalten. Es gibt sogar Funktionen der Textverarbeitung bietet Kontrolle über Schriftarten, Fett oder kursiv-Schrift, Größe, Farbe und Seitenformatierung.

Bei Änderung der referenzierten Zellen einer Formel aktualisiert die Software den erst angezeigten Wert der Formelzelle normalerweise automatisch, ggf. aber auch nur auf Anforderung. Werden Formeln eines Tabellenfeldes an andere Stellen kopiert, muss zwischen absolutem und relativem Zellbezug unterschieden werden. Formelzellen können auf andere Formelzellen verweisen. Mit diesem Prinzip können komplizierte Rechengänge mit vielen verknüpften Teil-Ergebnissen übersichtlich dargestellt werden.

Verschiedene Arten des Einsatzes von Tabellenkalkulationsprogrammen können sein:

- Automatisches Ausführen und Aktualisieren von Berechnungen
- Datenspeicherung & Datenverwaltung
- Automatisches erstellen und aktualisieren von Diagrammen
- Modellierung
- Programmierung (Makros, Visual Basic)

Einsatzmöglichkeiten

Vor allem aus dem kaufmännischen Bereich ist sie nicht mehr wegzudenken, wie ein Auszug aus den Anwendungsfeldern in der mittleren Spalte zeigt. Aber auch im wissenschaftlichen Bereich hat Tabellenkalkulation ihren Platz, insbesondere dann, wenn ein spezifischer Funktionsvorrat für Standardaufgaben aus den jeweiligen Bereichen angeboten wird (zB Statistikfunktionen, finanzmathematische Funktionen)

privat	betrieblich	wissenschaftlich
Haushaltsbudget	Kalkulationen	Statistik
Angebotsvergleich	einfache Kostenrechnung	Charts
Kostenvergleich	Reisekostenabrechnung	mathematische Modelle
Kreditkostenberechnung	Statistik	kaufmännische Anwendungen
uvm...	Charts	volkswirtschaftliche Modelle
	Investitionsrechnung	...uvm
	einfache Fakturierung	
	ABC-Analyse	
	Budgetierung	

Einschränkungen

Jedoch ist auch ein Tabellenkalkulationsprogramm keine „Eierlegende Wollmichsau“ und man stößt auch hier an Grenzen:

- Datenverwaltung wird schnell aufwendig ⇒ Lösung: Datenbanksystem verwenden
- Beschränkt in der Grösse ⇒ Lösung: Datenbanksystem verwenden
- Zusammenhänge sind nicht sichtbar ⇒ Lösung: Gute Dokumentation
- Gefahr von Nebenwirkungen (side effects) ist gross ⇒ Lösung: Sorgfältig arbeiten

Praxis

- [Einführung](#)
 - [Übungen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

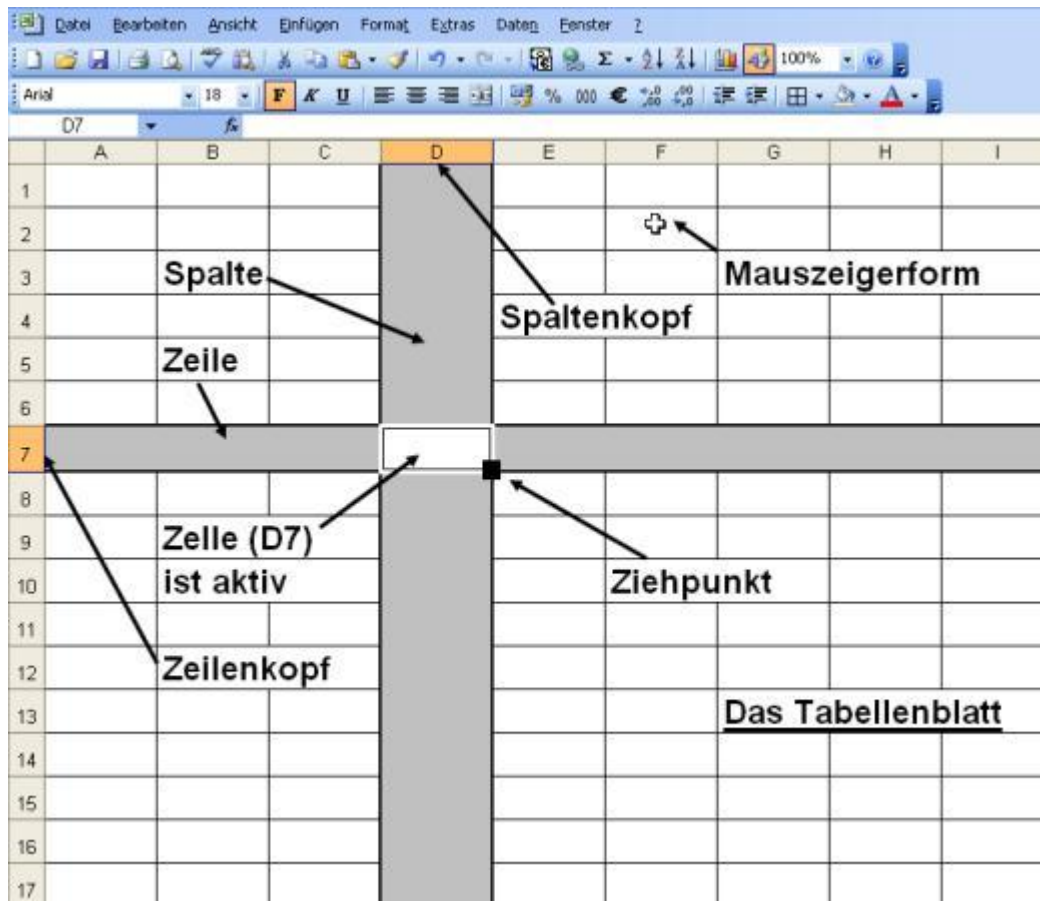
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:08



Last update: **2019/05/12 07:26**

Tabellenkalkulation - Einführung

Wichtige Begriffe



Zellenbezüge

- relativer Zellenbezug: A4; B12; KL77; AX500; ...
- absoluter Zellenbezug: \A\\$4; \\$B\\$12; \\$KL\\$77; \\$AX\\$500; ...
 - mittels 'F4' in Zelle hinzufügen.
- gemischter Zellenbezug: \\$A4; B\\$12; \\$KL77; AX\\$500; ...

Grundrechnungsarten

- Addition: =Zelle+Zelle
- Subtraktion: =Zelle-Zelle
- Multiplikation: =Zelle*Zelle
- Division: =Zelle/Zelle

Funktionen

SUMME

Die SUMME Funktion ist eine der meistgebräuchlichsten Funktionen in der Tabellenkalkulation, durch die gegebene Werte aufsummiert werden können. Dabei ist das Symbol Σ hilfreich.

```
=SUMME (BEREICHSANFANG:BEREICHSENDE)
```

Beispiele:

```
=SUMME (A1:A44)
```

```
=SUMME (B2:D20)
```

MITTELWERT

Mit dieser Funktion wird der Mittelwert des gewünschten Bereiches ausgegeben.

```
=MITTELWERT (BEREICHSANFANG:BEREICHSENDE)
```

Beispiele:

```
=MITTELWERT (A1:A44)
```

```
=MITTELWERT (B2:D20)
```

MIN / MAX

Während mit MIN() der geringste Wert eines Bereiches ausgegeben wird, sucht MAX() nach dem größten Wert.

```
=MIN (BEREICHSANFANG:BEREICHSENDE)
```

```
=MAX (BEREICHSANFANG:BEREICHSENDE)
```

Beispiele:

```
=MIN (A1:A44)
```

```
=MAX (B2:D20)
```

ANZAHL / ANZAHL2

Beide Funktionen berechnen, wie viele Werte in einem Bereich enthalten sind. Mit ANZAHL2() werden jedoch nur jene ausgegeben, die auch tatsächlich Werte enthalten.

```
=ANZAHL (BEREICHSANFANG:BEREICHSENDE)
```

```
=ANZAHL2 (BEREICHSANFANG:BEREICHSENDE)
```

Beispiele:

```
=ANZAHL (A1:A44)
```

```
=ANZAHL2(B2:D20)
```

WENN

Hierbei können Ausgaben an Bedingungen geknüpft werden. WENN das eintritt, DANN soll das passieren, SONST das andere. Deshalb wird sie auch als „Wenn-Dann-Sonst“ Funktion bezeichnet. Diese Funktion kommt einer IF Schleife in C++, oder anderen Programmiersprachen sehr nahe.

```
=WENN(BEDINGUNG;DANN;SONST)
```

Beispiel:

Wenn die Höhe des Hauses kleiner als 20 Meter ist,
soll die Zelle HAUS ausgeben, sonst HOCHHAUS!

```
=WENN(B4<10;"HAUS";"HOCHHAUS")
```

SVERWEIS

Der SVERWEIS sucht in einem angegebenen Zellbereich nach einem passenden Wert und holt diesen in ein bestimmtes Feld.

```
=SVERWEIS(SUCHWERT;BEREICHSTABELLE;ZIELSPALTE)
```

Beispiel:

Suche in einer Posttariftablette den richtigen Preis heraus

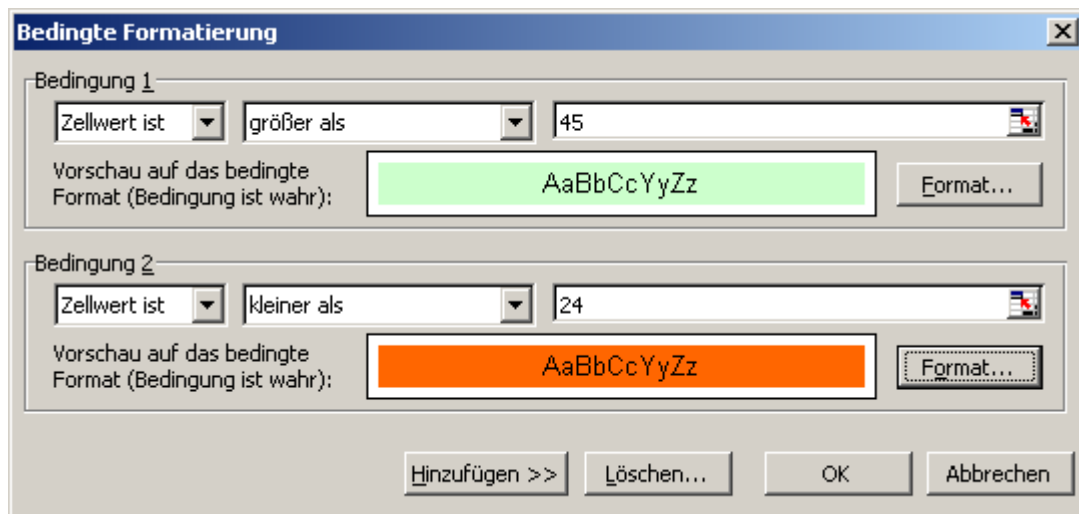
```
=SVERWEIS(B11;B5:C9;2)
```

	A	B	C
1	Die neuen Postgebühren		
2			
3	Briefe	Gewicht ab	Preis/Brief
4	Standard	0	0,62
5	Standard Plus	20	0,90
6	Maxi	50	1,45
7	Maxi Plus	500	2,90
8	Großbrief	1000	3,80
9		2000	
10			
11	Bitte Gewicht eingeben:	25	
12	Preis:	0,90	
13			

Grundfunktionen

Bedingte Formatierung

Das Format einer Zelle lässt sich durch bedingte Formatierung je nach dem Wert in der Zelle verändern:



Sortieren

Mit Hilfe des Sortierens können die Tabelleninhalte in eine gewünschte Reihenfolge gebracht werden. Dabei darf nur derjenige Bereich markiert werden, der auch tatsächlich sortiert werden möchte! Danach auf:

DATEN - Sortieren - Sortieren nach
und das gewünschte Suchkriterium auswählen

Filtern

Mit dieser Funktion können Inhalte einer Tabelle herausgefiltert werden. Erste Zeile eines Bereiches markieren und dann:

DATEN - Filter - AutoFilter

Inhalte einfügen

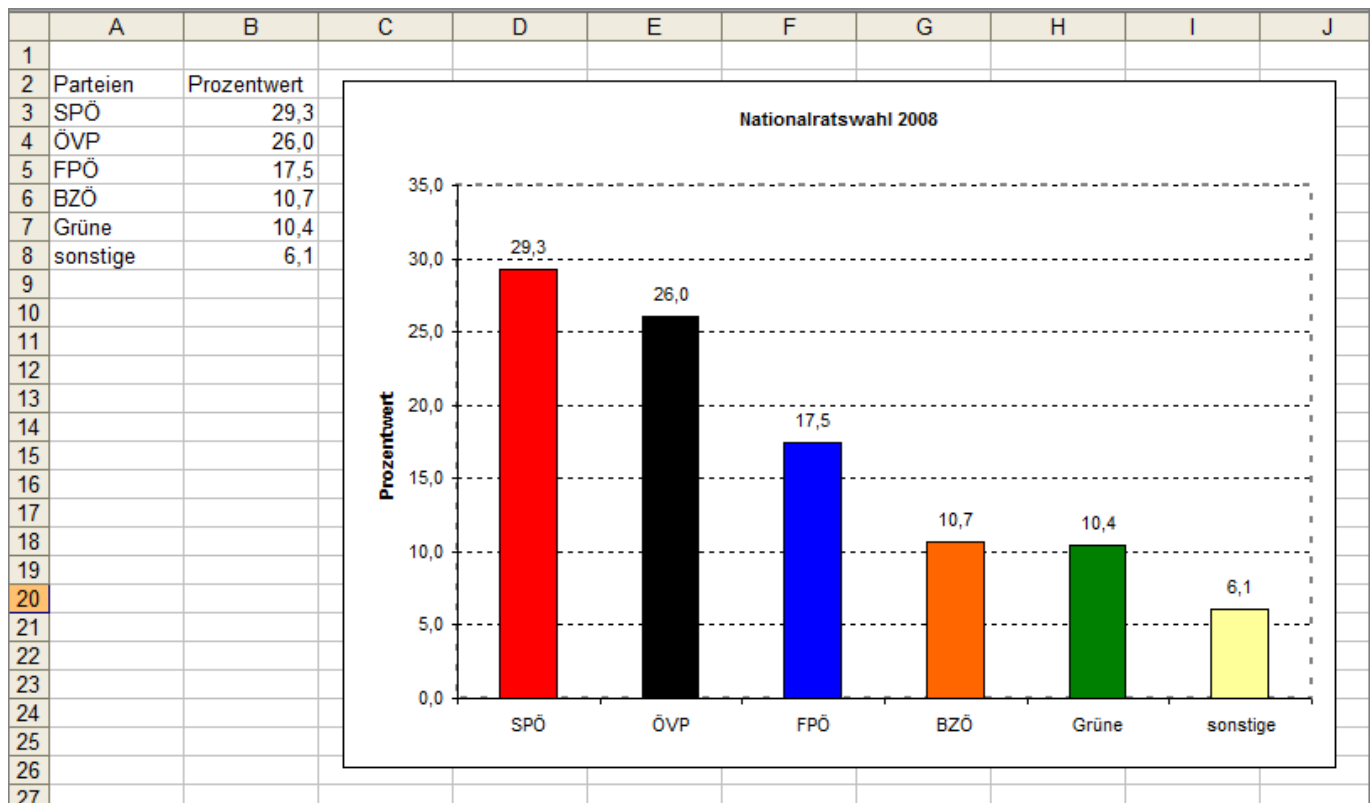
Mit dieser Funktion kann das Einfügen eines Textes zielführend beeinflusst werden. Einen Bereich markieren - kopieren - und an gewünschter Stelle:

BEARBEITEN - Inhalte einfügen
und dort kann die gewünschte Einfügeoption gewählt werden

Diagramm

Erstellung eines Diagramms:

- Tabelle erstellen (auf die Wahl von Zelle und Spalte sollte geachtet werden)
- EINFÜGEN - Diagramm
- einen passenden und für den Zweck geeigneten Diagrammtyp auswählen
- den Datenbereich festlegen
- Einstellungen für den Titel, der Achsen, der Legende, ... vornehmen
- Ort zum Einfügen des Diagramms festlegen - FERTIG STELLEN
- Jetzt kann das Diagramm noch formatiert (Schriftgröße, Hintergrund, ...) werden, indem das zuändernde Objekt markiert wird und RECHTE MAUSTASTE gedrückt wird.



From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:08:08_01



Last update: 2019/05/12 06:44

Übungsaufgaben

1. Klassenübersicht

Gestalte ein Excelworksheet für einen Lehrer, der sich in Informatik nicht auskennt.

Der Lehrer hat Schularbeiten, Mitarbeitskontrollen und Hausübungen.

Schularbeiten haben immer 48 Punkte, Mitarbeitskontrollen können unterschiedliche Gesamtpunkte haben, Hausübungen haben 0 bis 5 Punkte.

Jeder dieser drei Bereiche wird auf einem eigenen Tabellenblatt dargestellt.

Der Mittelwert jeweils aller Schularbeiten, Mitarbeitskontrollen und Hausübungen wird im 48-Punkte-Schema ausgegeben.

Auf einem eigenen Tabellenblatt wird die Semesternote ermittelt. Sie berechnet sich wie folgt: 40% Schularbeiten, 30% Mitarbeitskontrollen und 30% Hausübungen.

Verwende die bedingte Formatierung: Eine negative Note soll einen roten Hintergrund erhalten, ein „Sehr gut“ soll grün hinterlegt werden.

Stelle die Notenübersicht mittels Balkendiagramm dar!

2. Excelübung 01

- [excel_uebung01.rtf](#)

3. Excelübung 02

- [excel_uebung02.rtf](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:08:08_01:08_01_01

Last update: **2019/05/12 06:45**



2) DATENBANKEN

- 2.1) Allgemeines
- 2.2) Datenmodellierung
- 2.3) Entity Relationship Modell (Konzeptionelles Modell)
 - 2.3.1) Übungen
- 2.4) Relationenmodell (Logisches Modell)
 - 2.4.1) Übungen
- 2.5) Umsetzung ER-Modell --> Relationenmodell
 - 2.5.1) Übungen
- 2.6) Normalformen
- 2.7) SQL (Physisches Modell)
- 2.8) SQL Anbindung mit PHP
 - 2.8.1) Übung
 - SQL-ISLAND GAME - Schaffst du es den Piloten zu befreien?
 - SOLOLEARN - Play with SQL

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:09



Last update: **2019/05/04 09:24**

[Fragen](#) **Inhalte** [Arbeitsaufträge](#)

Das Satzsystem La-TeX

- [LaTeX - Was ist das?](#)
- [Rund um LaTeX](#)
- [Installationsanleitung](#)
- [Einführungen und Dokumentationen](#)
- [Literaturverzeichnis mit BibTeX](#)
- [Generierung von Literatureinträgen](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:10



Last update: **2019/05/04 09:25**

- [HTML 5-CSS](#)
- [CSS-Kurs](#)
- [Javascript](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:11



Last update: **2019/05/06 19:34**

Interaktion und Dynamik mit JavaScript

- [Einführung](#)
- [Hintergründe](#)
- [Einfache Befehle](#)
- [Grundlegende Sprachelemente](#)
- [Kontrollstrukturen](#)
- [Funktionen](#)
- [Objekte](#)
- [Vordefinierte Objekte](#)
- [Objektmodell](#)
- [Zugriff auf HTML-Dokumente](#)
- [Event-Handler](#)
- [Unsere Scripts](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:11:11_03



Last update: **2019/05/06 19:49**

1) PHP

1.1) Was ist PHP?

- [1.1.1\) Was ist PHP](#)
- [1.1.2\) How to use PHP](#)

1.2) Grundlegende Sprachelemente

- [1.2.1\) Kommentare](#)
- [1.2.2\) Ausgabe](#)
- [1.2.3\) ÜBUNG 1](#)

1.3) Variablen und Operatoren

- [1.3.1\) Variablen und Operatoren](#)

1.4) Interaktive Webseiten

- [1.4.1\) Formulare](#)
- [1.4.2\) ÜBUNG 2](#)

1.5) Anführungszeichen

- [1.5.1\) Verwenden von Anführungszeichen](#)

1.6) Kontrollstrukturen

- [1.6.1\) Vergleichsoperatoren](#)
- [1.6.2\) if-Anweisung](#)
- [1.6.3\) switch-Anweisung](#)
- [1.6.4\) ÜBUNG 3-7](#)

1.7) Schleifen

- [1.7.1\) for-Schleife](#)
- [1.7.2\) while-Schleife](#)
- [1.7.3\) do-while-Schleife](#)
- [1.7.4\) foreach-Schleife](#)

- [1.7.5\) ÜBUNG 8-11](#)

1.8) Felder

- [1.8.1\) Grundlagen zu Felder](#)
- [1.8.2\) Indizierte Felder](#)
- [1.8.3\) Assoziative Felder](#)
- [1.8.4\) Mehrdimensionale Felder](#)
- [1.8.5\) Weitere Informationen zu Feldern](#)
- [1.8.6\) ÜBUNG 12-18](#)
- [1.8.7\) KONTROLLFRAGEN](#)

1.9) Formulare

- [1.9.1\) Hidden-Feld](#)
- [1.9.2\) Textarea](#)
- [1.9.3\) Checkbox](#)
- [1.9.4\) Radio-Button](#)
- [1.9.5\) Auswahllisten](#)
- [1.9.6\) ÜBUNG 20](#)

1.10) Externe Dateien

- [1.10.1\) Theorie: Nutzung von externen Dateien](#)
- [1.10.2\) Dateien öffnen, lesen und schließen](#)
- [1.10.3\) Aus Dateien lesen](#)
- [1.10.4\) In Dateien schreiben](#)
- [1.10.5\) ÜBUNG 21 Gästebuch](#)
- [1.10.6\) ÜBUNG 22 Besucher zählen](#)
- [1.10.7\) ÜBUNG 23-25](#)

1.11) Datum und Zeit

- [1.11.1\) Datum und Zeit](#)
- [1.11.2\) ÜBUNG 26](#)

1.12) Funktionen

- [1.12.1\) Funktionen erstellen und aufrufen](#)
- [1.12.2\) Funktionen verwenden](#)
- [1.12.3\) Gültigkeitsbereich von Variablen](#)
- [1.12.4\) PHP-Dateien einbinden](#)

- [1.12.5\) Andere Dateitypen einbinden](#)
 - [1.12.6\) ÜBUNG 27-29](#)
-

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

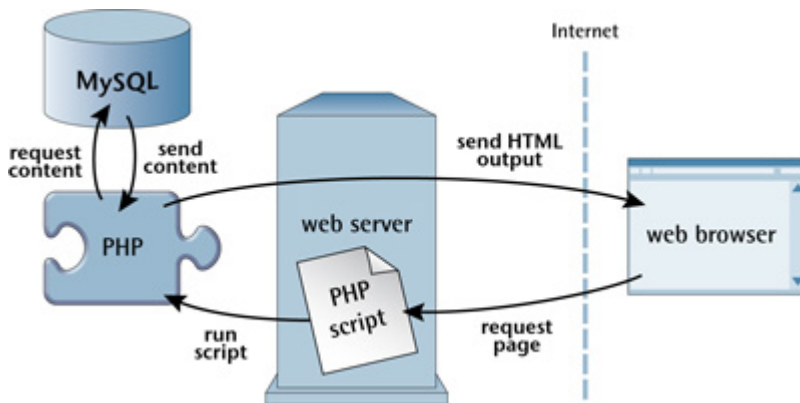
Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:12



Last update: **2019/05/04 09:36**

SQL-Anbindung via PHP



- 2.8) SQL Anbindung mit PHP
 - 2.8.1) Übung

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

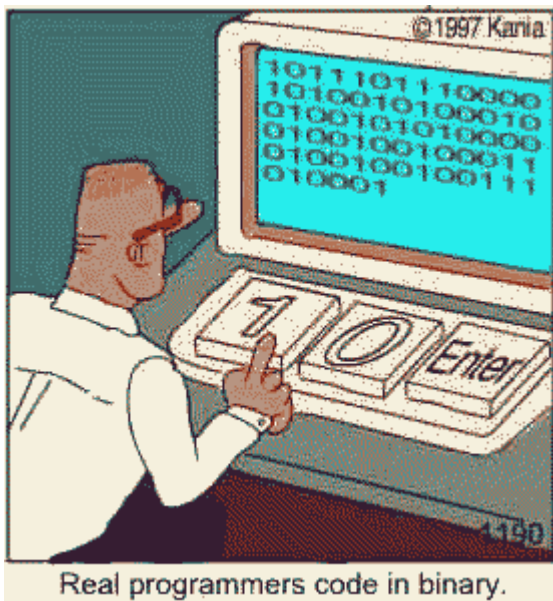
Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:13



Last update: **2019/05/10 14:47**

6) Algorithmik und Programmierung



- 6.1) Grundlagen zur Programmierung
- 6.2) Programmiersprachen
- 6.3) Compiler & Interpreter
- 6.4) Programmierstile
- 6.5) Visualisierung der Programmlogik/von Algorithmen
- 6.6) Code.org
- 6.7) Scratch
- 6.8) C++

From:

<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:14



Last update: **2019/05/04 09:38**

Grundlagen des (objektorientierten) visuellen Programmierens

Theorie

Objektorientiertes Programmieren

- [OOP - Klassen](#)

Programme mit grafischer Benutzeroberfläche

GUI

Grafische Benutzeroberfläche oder auch grafische Benutzerschnittstelle (Abk. GUI von englisch graphical user interface) bezeichnet eine Form von Benutzerschnittstelle eines Computers. Sie hat die Aufgabe, Anwendungssoftware auf einem Rechner mittels grafischer Symbole, Steuerelemente oder auch Widgets genannt, bedienbar zu machen. Dies geschieht bei Computern meistens mittels einer Maus als Steuergerät, mit der die grafischen Elemente bedient oder ausgewählt werden, bei Smartphones, Tablets und Kiosksystemen in der Regel durch Berührung eines Sensorbildschirms.

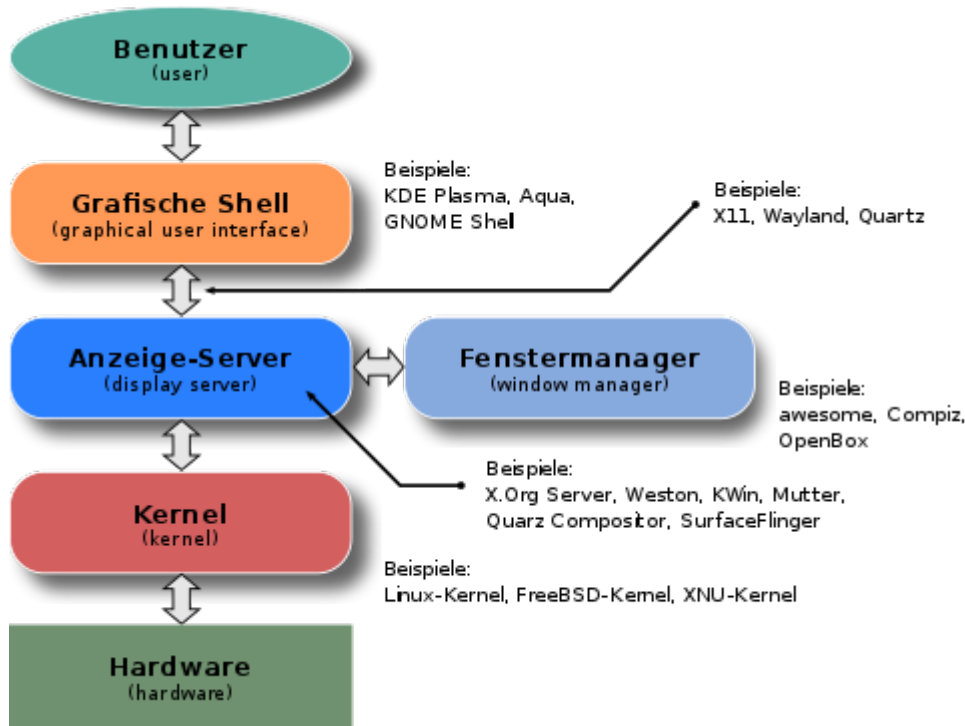
Fenstersystem

Ein Fenstersystem (englisch windowing system) ist der Unterbau einer grafischen Benutzeroberfläche (GUI), deren Hauptaufgabe die Verwaltung von Programmfenstern ist. Im Normalfall ist es Teil einer größeren Desktop-Umgebung.

Aus der Sicht eines Programmierers implementiert das Fenstersystem die grafischen Basisfunktionen, wie das Darstellen von Schriftarten, Zeichnen von Linien, Kurven und Pixelgrafiken, und das Abstrahieren der Grafikhardware (Grafikkarte).

Das Fenstersystem gestattet es dem Anwender mit mehreren Programmen gleichzeitig zu arbeiten, indem jedes Programm „in“ einem oder mehreren eigenen Bereichen des Bildschirms, den Fenstern, ausgeführt wird, die üblicherweise rechteckig sind, mit dem Zeigegerät (Maus) frei bewegt werden können und einander überlappen dürfen.

Einige Fenstersysteme, wie das X Window System in Unix-artigen Umgebungen, haben erweiterte Fähigkeiten wie Netzwerktransparenz, die es dem Anwender gestatten die grafische Oberfläche einer Anwendung auf einem anderen Computer darzustellen. Das X-Window-System implementiert auch kein festes Aussehen der Umgebung, wodurch die Fenstermanager, GUI-Toolkits und Desktop-Umgebungen volle Freiheit bei der optischen Gestaltung und Handhabung haben.



GUI in C++

C++ bringt von Haus aus kein GUI mit, da es als hardwarenahe plattformunabhängige Sprache erstellt wurde und GUIs stark vom verwendeten Betriebssystem abhängen. Dieser „Mangel“ könnte einige Neueinsteiger vielleicht auf den ersten Blick abschrecken, da sie sich einen Einstieg in C++ oder gar in die gesamte Programmierung mit vielen Buttons, Textfeldern, Statusanzeigen und Menüeinträgen erhofft haben.

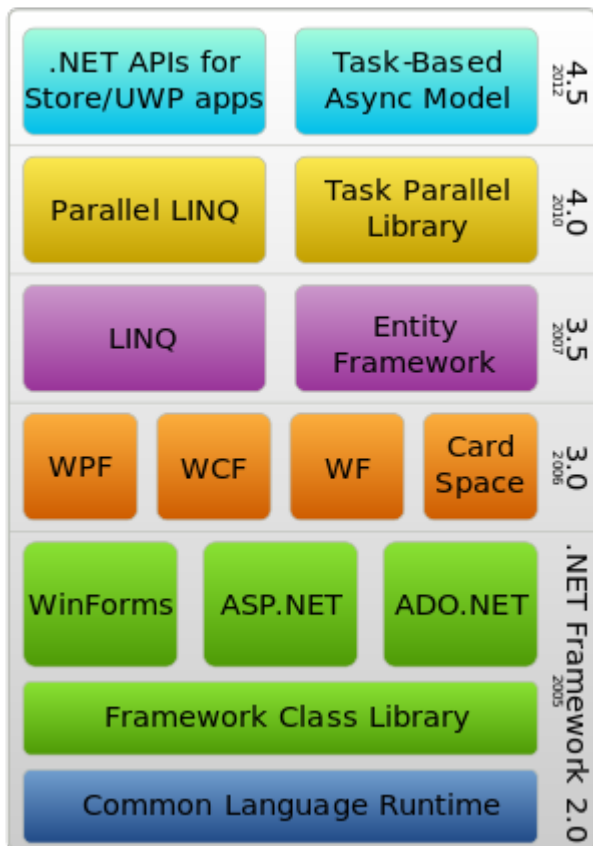
Sobald man mit C++ dann doch ein GUI programmieren will, sollte auf externe Bibliotheken zurückgegriffen werden. Alle GUIs erfordern ein Verständnis der C++ Grundlagen. Nachfolgend sind einige Bibliotheken zur GUI-Programmierung aufgelistet. Viele der Bibliotheken lassen sich auch mit anderen Programmiersprachen einsetzen.

.NET-Framework

.NET Framework (auch mit dotNetFx oder nur mit NetFx abgekürzt) ist ein **Teil von Microsofts Software-Plattform .NET**. Es umfasst eine Sammlung zahlreicher **Werkzeuge und Schnittstellen**, die die **Entwicklung von Anwendungsprogrammen unterstützen und vereinfachen** sollen, sowie eine Laufzeitumgebung für Programme, die mithilfe dieses Frameworks geschrieben wurden. Da diese Programme hierbei nicht herkömmlich direkt in Instruktionen für einen Prozessor übersetzt werden, braucht ein System für die Ausführung der Programme zusätzliche Software (welche die Laufzeitumgebung des Frameworks bildet). Programme, die mit dem .NET Framework geschrieben wurden, werden in der Regel so ausgeliefert, dass benötigte Komponenten des Frameworks automatisch mitinstalliert werden. Der Benutzer muss sich also nur im Ausnahmefall um das Vorhandensein der Laufzeitumgebung kümmern. Außerdem ist das Framework seit vielen Jahren bereits Bestandteil von Windows Installationen, wenn auch nicht immer in der aktuellsten Version und daher ist diese für manche Zwecke nicht ausreichend.

Ziele des Frameworks, wie auch der Plattform .NET, sind unter anderem **Kompatibilität zwischen**

verschiedenen Plattformen, Zusammenführung bisher inkompatibler Microsoft Technologien und Verbesserung veralteter Standards. Aus Anwendersicht birgt das Framework zwar keine direkten Vorteile, da sich seine Funktionen und Anwendungsbereiche hauptsächlich auf Softwareentwicklung, -betrieb und -wartung beschränken, dennoch ist es eine notwendige Voraussetzung für den Betrieb zahlreicher Anwendungsprogramme.



Qt

Qt ist eine leistungsstarke plattformübergreifende Klassenbibliothek mit Meta-Object Compiler (gennant: moc), die von der norwegischen Firma Trolltech entwickelt wurde. Trolltech wurde 2008 von Nokia übernommen, welches jedoch wirtschaftliche Schwierigkeiten hatte und in der Folge Qt 2011 an das finnische Softwareunternehmen Digia verkaufte.

Die Entwicklung wird inzwischen vom Qt Project vorangetrieben, so dass jeder sich daran beteiligen kann. Die Klassenbibliothek ist unter der GNU General Public License (GPL) und der GNU Lesser General Public License (LGPL) lizenziert. Es gibt außerdem eine proprietäre Lizenz von Digia, die allerdings lediglich zusätzlichen technischen Support beinhaltet.

Speziell aufgrund der Veränderungen, die mit Qt 5 und C++ Einzug in das Framework halten, ist Qt eine der mächtigsten Bibliotheken für C++. Ein weiterer Vorteil von Qt ist die vollständig freie Entwicklungsumgebung Qt Creator, die insbesondere Anfängern das Programmieren leichter macht.

VCL - Visual Component Library

Die **Visual Component Library (VCL)** ist ein GUI-Toolkit für Windows-Anwendungen. Sie wurde von

Embarcadero (Rad Studio), vormals Borland, Inprise und CodeGear, erstellt. Die VCL kann in den Programmiersprachen Borland Delphi, C + +, C, C# verwendet werden. Sie wird von den meisten Borland-Entwicklungsumgebungen als Komponentensammlung benutzt.

MFC - Microsoft Foundation Classes

Die Microsoft Foundation Classes (MFC) sind eine Sammlung objektorientierter Klassenbibliotheken (GUI-Toolkit), die von Microsoft für die Programmierung von Anwendungen mit grafischen Benutzeroberflächen für Windows mit C + + entwickelt wurden.

WF - Windows Forms

Windows Forms ist ein **GUI-Toolkit des Microsoft .NET Frameworks**. Es ermöglicht die Erstellung grafischer Benutzeroberflächen (GUIs) für Windows in C#. Im Vergleich zu Microsoft Foundation Classes (MFC), die auf der Programmiersprache C + + basiert, ist der Einstieg in die Programmierung mit Windows Forms einfacher. Das Framework basiert nicht auf dem Paradigma Model View Controller (MVC).

Mit .NET Framework 3.0 wurde von Microsoft eine Alternative zu Windows Forms bereitgestellt, die Windows Presentation Foundation, welche eine stärkere Trennung der grafischen Oberfläche vom Programmcode und – unter Zuhilfenahme von XAML, einer XML-basierenden Sprache – ein dynamischeres Layout ermöglicht.

GUI in C#

WPF - Windows Presentation Foundation

Windows Presentation Foundation (kurz WPF), auch bekannt unter dem Codenamen Avalon, ist ein Grafik-Framework und Fenstersystem des .NET Frameworks von Microsoft. Es wird seit Windows Vista mit Windows ausgeliefert und lässt sich auf Windows XP (bis zur Version 4.0) und Windows Server 2003 nachinstallieren. Für das neue Framework .NET Core soll WPF ab der Version 3.0 (für 2019 erwartet) unter Windows zur Verfügung stehen.[2]

WPF ist eine 2006 neu eingeführte Klassenbibliothek, die zur Gestaltung von grafischen Benutzeroberflächen und zur Integration von Multimedia-Komponenten und Animationen dient. Sie vereint DirectX, Windows Forms, Adobe Flash, HTML und CSS.

WPF stellt ein umfangreiches Modell für den Programmierer bereit. Dabei werden die Präsentation und die Geschäftslogik getrennt, dies wird vor allem durch die Auszeichnungssprache XAML (basierend auf XML) unterstützt. XAML beschreibt Oberflächen-Hierarchien deklarativ als XML-Code. WPF-Anwendungen können sowohl Desktop- als auch Web-Anwendungen sein und benutzen, wenn möglich, Hardwarebeschleunigung. Das Framework versucht, die verschiedenen Bereiche, die für die Präsentation wichtig sind (Benutzerschnittstelle, Zeichnen und Grafiken, Audio und Video, Dokumente, Typographie), zu vereinen.

Praxis

Visual C ++

- [Kurzmeldungen und Kurzeingaben](#)
- [Listbox](#)
- [Memobox](#)
- [Strings](#)
- [RichEdit](#)
- [Datenstrukturen](#)
- [Zugriff auf Datenbanken](#)
- [Image-Komponente](#)
- [Vertiefende Übungen](#)

Links:

- [Turbo C++ Buch Inhaltsverzeichnis](#)
- [Kapitel 1](#)
- [Kapitel 2](#)
- [Kapitel 3,1-9](#)
- [Kapitel 3,10-19](#)
- [Kapitel 3,20-24](#)
- [Kapitel 4](#)
- [Kapitel 5](#)
- [Kapitel 6](#)
- [Kapitel 7](#)
- [Kapitel 8](#)
- [Kapitel 9](#)
- [Kapitel 10 visuelle Komponenten](#)

-
- [Entwicklungsumgebung für C++ \(RAD Studio XE\)](#)
 - [Entwicklungsumgebung für C++ \(Visual Studio 2013\)](#)
 - [APP-Programmierung \(unter RAD Studio XE7\)](#)
 - [Erstellen eines ausführbaren EXE-Projekts](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:15

Last update: **2019/05/12 07:58**



Objektorientierte Programmierung (OOP)

Am Anfang steht wohl die Frage was Objektorientierte Programmierung überhaupt ist. Dies ist nicht ganz einfach zu erklären und es gibt viele Definitionen die dies versuchen.

Grob gesagt ist **Objektorientierte Programmierung (OOP) ein Verfahren zur Strukturierung von Programmen, bei dem Programmlogik zusammen mit Zustandsinformationen (Datenstrukturen) in Einheiten, den Objekten, zusammengefasst werden.**

Es ist also möglich mehrere Objekte des gleichen Typs zu haben. Jedes dieser Objekte hat dann seinen individuellen Zustand. Dies bietet die Möglichkeit einer besseren Modularisierung der Programme, sowie einer höheren Wartbarkeit des Quellcodes.

Klassen

Die Klasse (class) ist die zentrale Datenstruktur in C + +. Sie kapselt zusammengehörige Daten und Funktionen vom Rest des Programmes ab. Sie ist das Herz der objektorientierten Programmierung (OOP).

Klassen sind ein erweitertes Konzept von Datenstrukturen denen es erlaubt ist, außer Daten, noch Methoden zu beinhalten. Ein Objekt ist eine Instanz einer Klasse, welches sich die Eigenschaften der Klasse zunutze machen kann. Es ist möglich mehrere Objekte einer Klasse zu instanziiieren, wobei jedes dieser Objekte zwar die selben Eigenschaften hat, intern aber einen ganz individuellen Zustand haben kann.

Beispiel: Zwei Instanzen der Klasse „Gegner“ werden erzeugt. Nachdem ein Gegner Schaden erlitten hat sind seine Lebenspunkte auf 50 gesunken. Die des anderen Gegners haben allerdings noch den Wert 100.

Klassen werden normalerweise mit dem Schlüsselwort **class** deklariert. Außerdem haben sie immer folgendes Format:

```
class Klassenname
{
  Zugriffsspezifizierung 1:
  Member 1;
  ...
  Zugriffsspezifizierung 2:
  Member 2;
  ...
  ...
};
```

Der Klassenname identifiziert die Klasse, der Objektname ist eine optionale Liste von Namen von Objekten dieser Klasse. Der Körper der Klasse wird durch geschweifte Klammern gekennzeichnet und kann verschiedene Member (Methoden, Membervariablen) beinhalten. Diesen können verschiedene Zugriffsspezifizierungen zugewiesen werden.

Zugriffsrechte

public: Auf Members kann von überall zugegriffen werden von wo das Objekt sichtbar ist

private: Auf Member kann nur innerhalb anderer Member zugegriffen werden oder aus befreundeten Klassen und Funktionen

protected: Members sind verfügbar für Member der selben Klasse, befreundeten Funktionen/Klassen und Abgeleiteten Klassen

Attribute

Die Festlegung, welche Daten zu einem Typ gehören, erfolgt bei der Definition der Klasse. Die Objekte enthalten jeweils unabhängig voneinander einen Satz von Datenkomponenten (Attributen). Ihre Startwerte können durch Konstruktoren festgelegt werden. C++ erlaubt auch die Zuweisung von Anfangswerten bei der Definition:

```
class Enemy {  
    public:    //Zugriffsrecht public  
        void setHealth(int h);  
        int getHealth();  
        string name;  
        int id;  
    private: //Zugriffsrecht private  
        int health;  
};
```

Elementzugriff

Im Beispiel davor wird ein Objekt vom **Typ Enemy** angelegt. Das **Objekt** enthält die **3 Variablen health (integer), id (int) und name (String)**, wie es in der Klassendefinition zu sehen ist. Um auf eine öffentliche Elementvariable zugreifen zu können, wird an den Objektnamen ein **Punkt** und dann der in der Klassendefinition verwendete Elementname gehängt. Im Beispiel wird der Name auf Andreas gesetzt.

```
...  
//Zugriff auf öffentliche Variable name  
e1.name="Andreas";  
//Zugriff auf öffentliche Variable id  
e1.id=1;  
...  
//e1.health=100; ist nicht erlaubt, da health das Schlüsselwort private  
besitzt!!!
```

Klassenmethoden

Nun können Sie ein Objekt von der Klasse `Enemy` erzeugen. Aber was nützt Ihnen die schönste Datenstruktur in Ihrem Programm, wenn sie nicht durch Funktionen zum Leben erweckt wird? Sie werden z.B. einen Namen und Lebenspunkte eingeben und ausgeben wollen. Vielleicht wollen Sie die Lebenspunkte verringern oder erhöhen. Kurz gesagt, ein Datenverbund ist nichts wert ohne Funktionen, die auf ihn wirken. Aber die Funktionen sind auch nur im Zusammenhang mit ihrem Datenverbund sinnvoll. Aus diesem Grund werden die Funktionen ebenso in die Klasse integriert wie die Datenelemente. Eine Funktion, die zu einer Klasse gehört, nennt man Elementfunktion oder auf englisch *member function*. In anderen objektorientierten Programmiersprachen spricht man auch von einer Methode oder Operation. Aufruf So, wie Sie auf Datenelemente nur über ein real existierendes Objekt, also eine Variable dieser Klasse zugreifen können, kann auch eine Funktion nur über ein Objekt aufgerufen werden. Objekt und Funktionsnamen werden dabei durch einen Punkt getrennt. Die Funktion arbeitet mit den Daten des Objekts, über das sie gerufen wurde. Aus prozeduraler Sicht könnte man es so sehen, dass eine Elementfunktion immer bereits einen Parameter mit sich trägt, nämlich das Objekt, über das sie aufgerufen wurde.

Beispiel:

```
#include <iostream>
#include <conio.h>
#include <string.h>

using namespace std;

//Klasse Enemy
class Enemy {
public:    //Zugriffsrecht public
    void setHealth(int h);
    int getHealth();
    string name;
    int id;
private: //Zugriffsrecht private
    int health;
};

//Methode setHealth zum Setzen der Lebenspunkte
void Enemy::setHealth(int h)
{
    health=h;
}

//Methode getHealth() zum Auslesen der Lebenspunkte
int Enemy::getHealth()
{
    return health;
}

//Hauptprogramm
int main()
{
    Enemy e1;    //Objekt e1 der Klasse Enemy wird erzeugt

    //Zugriff auf öffentliche Variable name
```

```
e1.name="Andreas";  
//Zugriff auf öffentliche Variable id  
e1.id=1;  
  
//Aufruf der Methode setHealth  
e1.setHealth( 100 );  
//Aufruf der Methode getHealth und Zugriff auf die öffentliche Variable  
name  
cout << "Der Gegner " << e1.name << " hat " << e1.getHealth() << "  
Lebenspunkte.\n";  
//Aufruf der Methode setHealth  
e1.setHealth( 50 );  
//Aufruf der Methode getHealth und Zugriff auf die öffentliche Variable  
name  
cout << "Der Gegner " << e1.name << " hat " << e1.getHealth() << "  
Lebenspunkte.\n";  
  
return 0;  
}
```

Methodendefinition innerhalb einer Klasse

Alternativ können Sie die Elementfunktion auch direkt in der Klasse definieren. Das wird leicht unübersichtlich, darum sollten Sie das nur bei sehr kurzen Funktionen tun.

```
#include <iostream>  
#include <conio.h>  
#include <string.h>  
  
using namespace std;  
  
class Enemy {  
public:      //Zugriffsrecht public  
           /****** INLINE - METHODENDEFINITION *****/  
           //Methode setHealth zum Setzen der Lebenspunkte  
void setHealth(int h)  
{  
    health=h;  
}  
           //Methode getHealth() zum Auslesen der Lebenspunkte  
int getHealth()  
{  
    return health;  
}  
  
    string name;  
    int id;  
  
private:   //Zugriffsrecht private
```

```

        int health;
    };

//Hauptprogramm
int main()
{
    Enemy e1;    //Objekt e1 der Klasse Enemy wird erzeugt

    //Zugriff auf öffentliche Variable name
    e1.name="Andreas"

    //Aufruf der Methode setHealth
    e1.setHealth( 100 );
    //Aufruf der Methode getHealth und Zugriff auf die öffentliche Variable
name
    cout << "Der Gegner " << e1.name << " hat " << e1.getHealth() << "
Lebenspunkte.\n";
    //Aufruf der Methode setHealth
    e1.setHealth( 50 );
    //Aufruf der Methode getHealth und Zugriff auf die öffentliche Variable
name
    cout << "Der Gegner " << e1.name << " hat " << e1.getHealth() << "
Lebenspunkte.\n";

    return 0;
}

```

Aufruf

So, wie Sie auf Datenelemente nur über ein real existierendes Objekt, also eine Variable dieser Klasse zugreifen können, kann auch eine Funktion **nur über ein Objekt aufgerufen werden**.

Objekt und **Funktionsnamen** werden dabei durch einen **Punkt** getrennt. Die Funktion arbeitet mit den Daten des Objekts, über das sie gerufen wurde. Aus prozeduraler Sicht könnte man es so sehen, dass eine Elementfunktion immer bereits einen Parameter mit sich trägt, nämlich das Objekt, über das sie aufgerufen wurde.

```

....
Enemy e1;    //Objekt e1 der Klasse Enemy wird erzeugt

//Aufruf der Methode setHealth
e1.setHealth( 100 );
....

```

Konstruktor

Die Elementfunktion, die beim Erzeugen eines Objekts aufgerufen wird, nennt man Konstruktor. In dieser Funktion können Sie dafür sorgen, dass alle Elemente des Objekts korrekt initialisiert sind. Der Konstruktor trägt immer den Namen der Klasse selbst und hat keinen Rückgabotyp, auch nicht void.

Der Standardkonstruktor hat keine Parameter.

```
class Enemy {
    public:          //Zugriffsrecht public

        /**** Konstruktor ***/
        Enemy()
        {
            health=0;
            name="";
            id=0;
        }

        string name;
        int id;

    private:        //Zugriffsrecht private
        int health;
};
```

Dekonstruktor

Im Falle einer Datumsklasse wäre es sinnvoll, dass der Konstruktor alle Elemente auf 0 setzt. Daran kann jede Elementfunktion leicht erkennen, dass das Datum noch nicht festgelegt wurde. Sie könnten alternativ das aktuelle Datum ermitteln und eintragen. Im Beispiel ist auch ein Destruktor definiert worden, obwohl er im Falle eines Datums keine Aufgabe hat.

```
class Enemy {
    public:          //Zugriffsrecht public

        /**** Destruktor ***/
        ~Enemy()
        {
            cout << "Ressourcen von " << name << " wurden
freigegeben!";
        }

        char name[50];
        int id;

    private:        //Zugriffsrecht private
        int health;
};
```

Überladen von Konstruktor

Konstruktoren können genauso überladen werden wie normale Funktionen auch. Es kann neben dem Standardkonstruktor auch mehrere weitere Konstruktoren mit verschiedenen Parametern geben. Der Compiler wird anhand der Aufrufparameter unterscheiden, welcher Konstruktor verwendet wird.

```
class Enemy {
    public:        //Zugriffsrecht public

        /**** Konstruktor ***/
        Enemy()
        {
            name="Max Mustermann";
        }
        Enemy(string n)
        {
            name=n;
        }
        Enemy(string n, int h)
        {
            name=n;
            health=h;
        }
        /**** Destruktor ***/
        ~Enemy()
        {
            cout << "Ressourcen von " << name << " wurden
freigegeben!";
        }

        char name[50];
        int id;

    private:      //Zugriffsrecht private
        int health;
};
```

Je nachdem wie viele Parameter beim Konstruktoraufruf übergeben werden, wird der passende Konstruktor ausgewählt. Existiert gar kein Konstruktor so wird vom Compiler ein leerer Konstruktor eingefügt.

```
int main (void)
{
    Enemy e;                //führt den Standardkonstruktor Enemy() aus
    -> name="Max Mustermann"
    Enemy e("Fritz Phantom"); //führt den 2. Konstruktor mit den
    passenden Argumenten aus -> name="Fritz Phantom"
    Enemy e("Hans Wurst", 100); //führt den 3. Konstruktor mit den
    passenden Argumenten aus -> name="Hans Wurst", health=100
}
```

```
    return 0;  
}
```

Vererbung

Die Informatik steckt voller schöner Analogien. So hat die objektorientierte Programmierung den Begriff der »Vererbung« eingeführt, wenn eine Klasse von einer anderen Klasse abgeleitet wird und deren Eigenschaften übernimmt.

Basisklasse

Eine Klasse kann als Basis zur Entwicklung einer neuen Klasse dienen, ohne dass ihr Code geändert werden muss. Dazu wird die neue Klasse definiert und dabei angegeben, dass sie eine abgeleitete Klasse der Basisklasse ist. Daraufhin gehören alle öffentlichen Elemente der Basisklasse auch zur neuen Klasse, ohne dass sie erneut deklariert werden müssen. Man sagt, die neue Klasse erbt die Eigenschaften der Basisklasse.

Spezialisierung

Durch die Elemente, die in der abgeleiteten Klasse definiert werden, wird die abgeleitete Klasse zu einem besonderen Fall der Basisklasse. Sie besitzt alle Eigenschaften der Basisklasse. Sie können neue Elemente hinzufügen. Am folgenden Beispiel wird deutlich, warum das Hinzufügen von Eigenschaften eine Spezialisierung ist.

Ein Beispiel

Aus Sicht eines Computerprogramms haben alle **Personen** (=Basisklasse)

- Namen
- Adressen und
- Telefonnummern.

Geschäftspartner haben darüber hinaus eine

- Bankverbindung.

Da die Geschäftspartner auch Personen sind, haben sie neben ihrer Bankverbindung auch Namen, Adressen und Telefonnummern (von der Basisklasse).

Einige Geschäftspartner können auch **Kunden** sein. **Kunden** haben zusätzlich zu den Eigenschaften eines Geschäftspartners noch eine

- Lieferanschrift.

Lieferanten sind keine Kunden, aber auch Geschäftspartner. Sie haben noch

- eine Anzahl an offenen Rechnungen.

Selbst **Mitarbeiter** sind eigentlich Geschäftspartner, denn sie haben eine Bankverbindung. Darüber

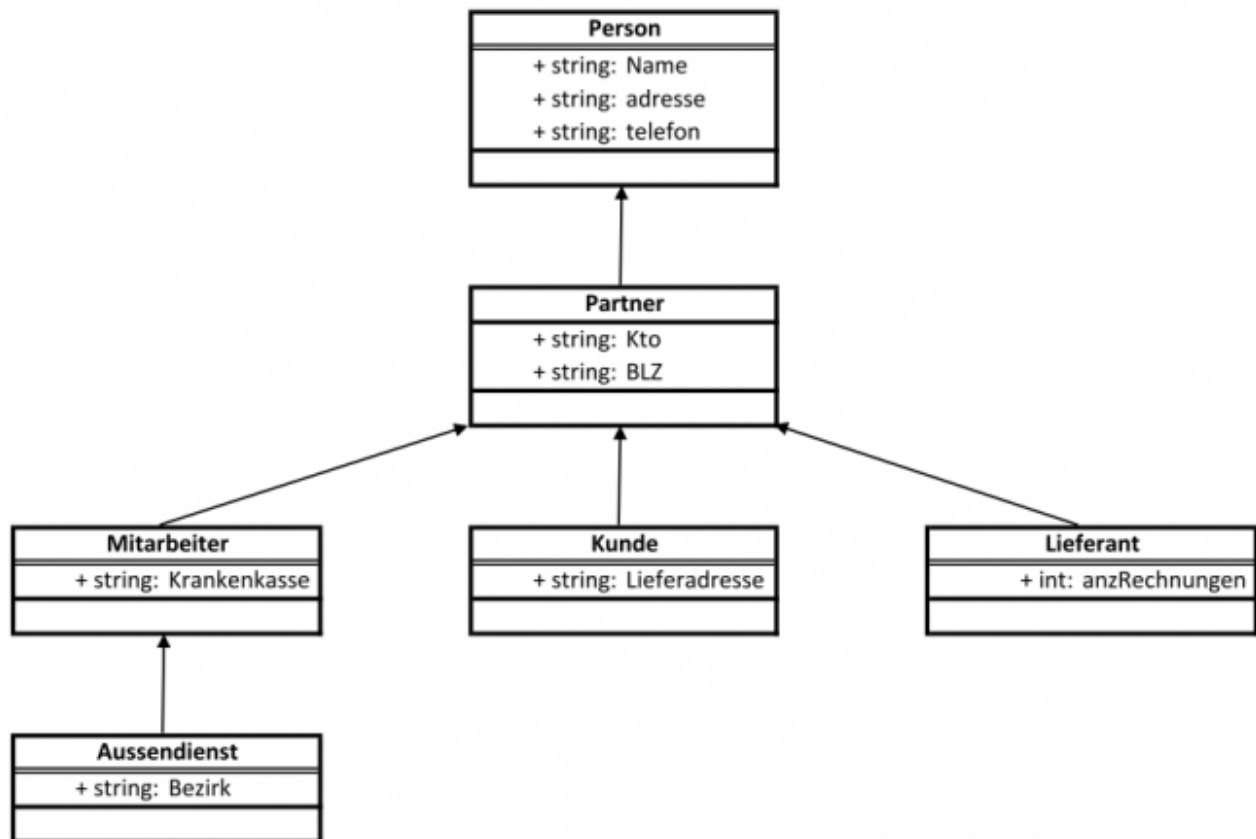
hinaus haben sie eine

- Krankenkasse.

Außendienstler haben alle Eigenschaften eines Mitarbeiters und zusätzlich ihren

- Bezirk.

Beispiel als UML-Diagramm:



Beispiel in C++

```

class Person
{
public:
    string Name, Adresse, Telefon;
};

class Partner : public Person
{
public:
    string Kto, BLZ;
};

class Mitarbeiter : public Partner
{
public:
    string Krankenkasse;
};
  
```

```
};

class Kunde : public Partner
{
    public:
        string Lieferadresse;
};

class Lieferant : public Partner
{
    public:
        int anzRechnungen;
};

class Aussendienst : public Mitarbeiter
{
    public:
        string Bezirk;
};

int main (void) {

    Aussendienst a1;

    a1.Bezirk="Amstetten";
    a1.Name="Hans Wurst";
    a1.Krankenkasse="Gebietskrankenkasse";
    a1.BLZ="14200";

    return 0;
}
```

Vorteil von Vererbungen:

Damit stellt die Ausgangsklasse Person die Verallgemeinerung dar und jede abgeleitete Klasse eine Spezialisierung. Der Vorteil dieser Technik ist, dass der bestehende Code der Basisklasse nicht noch einmal für die neu geschaffene Klasse geschrieben werden muss. Beispielsweise würde eine Prüffunktion der Adresse, die der Klasse Person hinzugefügt wird, automatisch auch allen anderen Klassen, die direkt oder indirekt von Person abgeleitet wurden, hinzugefügt, ohne dass eine Zeile Code mehr geschrieben werden müsste. Eine Änderung in der Klasse Mitarbeiter würde immer auch auf die Außendienstler durchschlagen. Es gibt aber keine Rückwirkung auf die Geschäftspartner.

Zusammenfassung

Die objektorientierte Programmierung hat einige neue Begriffe aufgebracht.

Objekt und Klasse

Der zentrale Begriff des Objekts bezeichnet einen Speicherbereich, der durch eine Klasse beschrieben wird. Prinzipiell kann sich der Anfänger ein Objekt als eine besondere Art einer Variablen vorstellen und die Klasse als die Typbeschreibung. Im Buch wird ein Objekt auch hin und wieder als Variable bezeichnet, insbesondere dann, wenn sich ein Objekt an dieser Stelle wie jede andere Variable verhält.

Attribut

Die Daten-Elemente einer Klasse werden in diesem Buch meist Elementvariable genannt. In der objektorientierten Literatur findet sich dafür auch die Bezeichnung Attribut. Damit wird angedeutet, dass die Elementvariablen die Eigenschaften eines Objekts beschreiben.

Methode und Operation

Die Funktionen einer Klasse, die hier als Elementfunktionen bezeichnet werden, finden sich in der objektorientierten Literatur unter dem Namen Methode oder Operation wieder. Diese Bezeichnung bringt zum Ausdruck, dass die Funktion nur über das Objekt erreichbar und damit eine Aktion des Objekts ist.

Konstruktor

Die Elementfunktion, die beim Erzeugen eines Objekts aufgerufen wird, nennt man Konstruktor. In dieser Funktion können Sie dafür sorgen, dass alle Elemente des Objekts korrekt initialisiert sind. Der Konstruktor trägt immer den Namen der Klasse selbst und hat keinen Rückgabetyp, auch nicht void. Der Standardkonstruktor hat keine Parameter.

Dekonstruktor

Im Falle einer Datumsklasse wäre es sinnvoll, dass der Konstruktor alle Elemente auf 0 setzt. Daran kann jede Elementfunktion leicht erkennen, dass das Datum noch nicht festgelegt wurde. Sie könnten alternativ das aktuelle Datum ermitteln und eintragen. Im Beispiel ist auch ein Destruktor definiert worden, obwohl er im Falle eines Datums keine Aufgabe hat

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:15:15_01

Last update: **2019/05/10 14:54**



C++

- [3.1\) Zeiger \(Wdhg. 7.Klasse\)](#)
 - [3.1.1\) Zeiger-Übung \(Wdhg. 7.Klasse\)](#)
- [3.2\) Rekursionen](#)
 - [3.2.1\) Rekursion-Übung](#)
- [3.3\) Datenstruktur struct](#)
- [3.4\) Einfach verkettete Listen](#)
- [3.5\) Doppeltverkettete Listen](#)
- [3.6\) Binäre Bäume](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:16



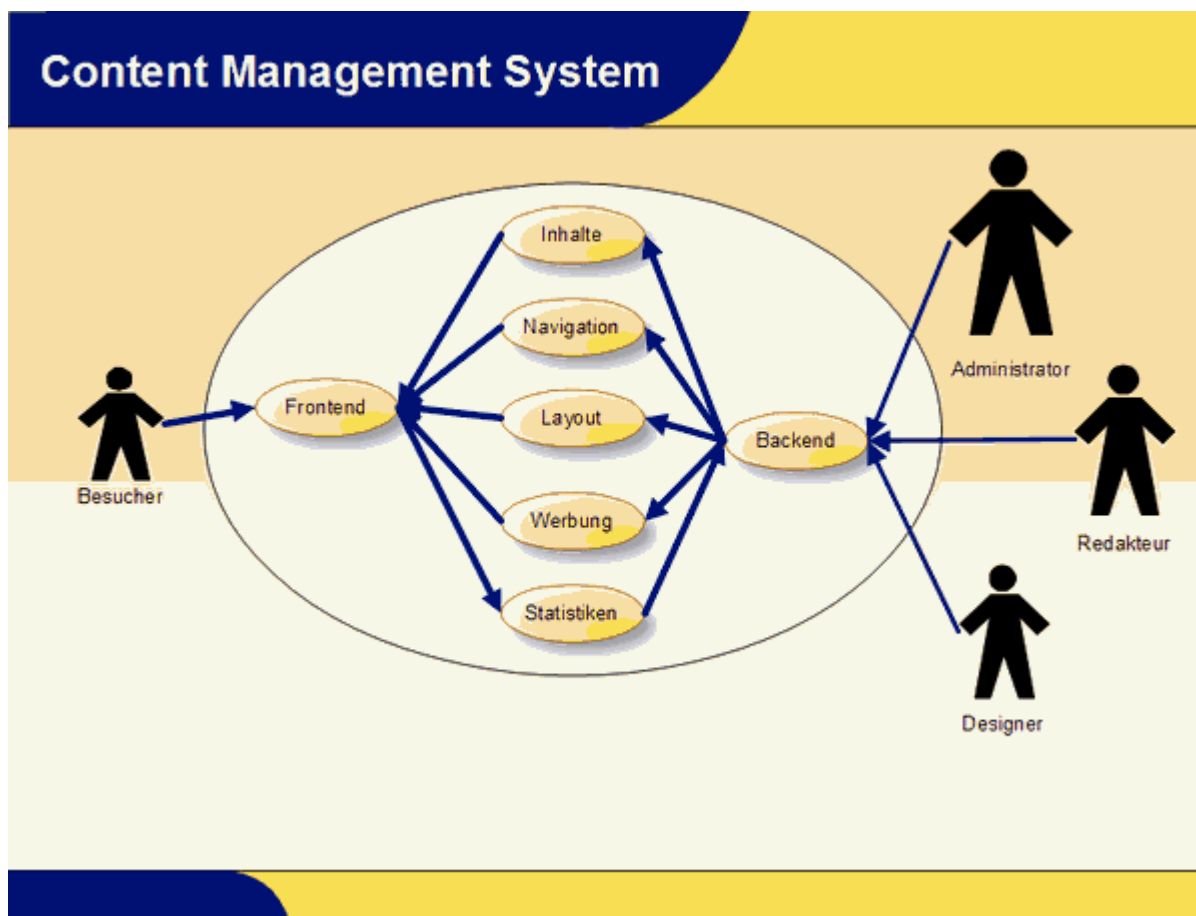
Last update: **2019/05/04 10:13**

Content Management System (CMS-System)

Ein CMS, ein **Content-Management-System**, dient zur **Verwaltung der Inhalte (Einpflanze von Texten, Bildern, Videos und Daten)**. Ein WCMS (Web Content Management System) verwaltet Inhalte von einer oder auch mehrerer **Websites**. WCMS sind **mandantenfähig**, d.h. sie können mehrere Kunden (Mandanten) bedienen, ohne dass diese Einblick in die Benutzerverwaltung des anderen haben. In weiterer Folge, wird immer von einem WCMS ausgegangen, jedoch der Einfachheit halber wird immer der Begriff CMS verwendet.

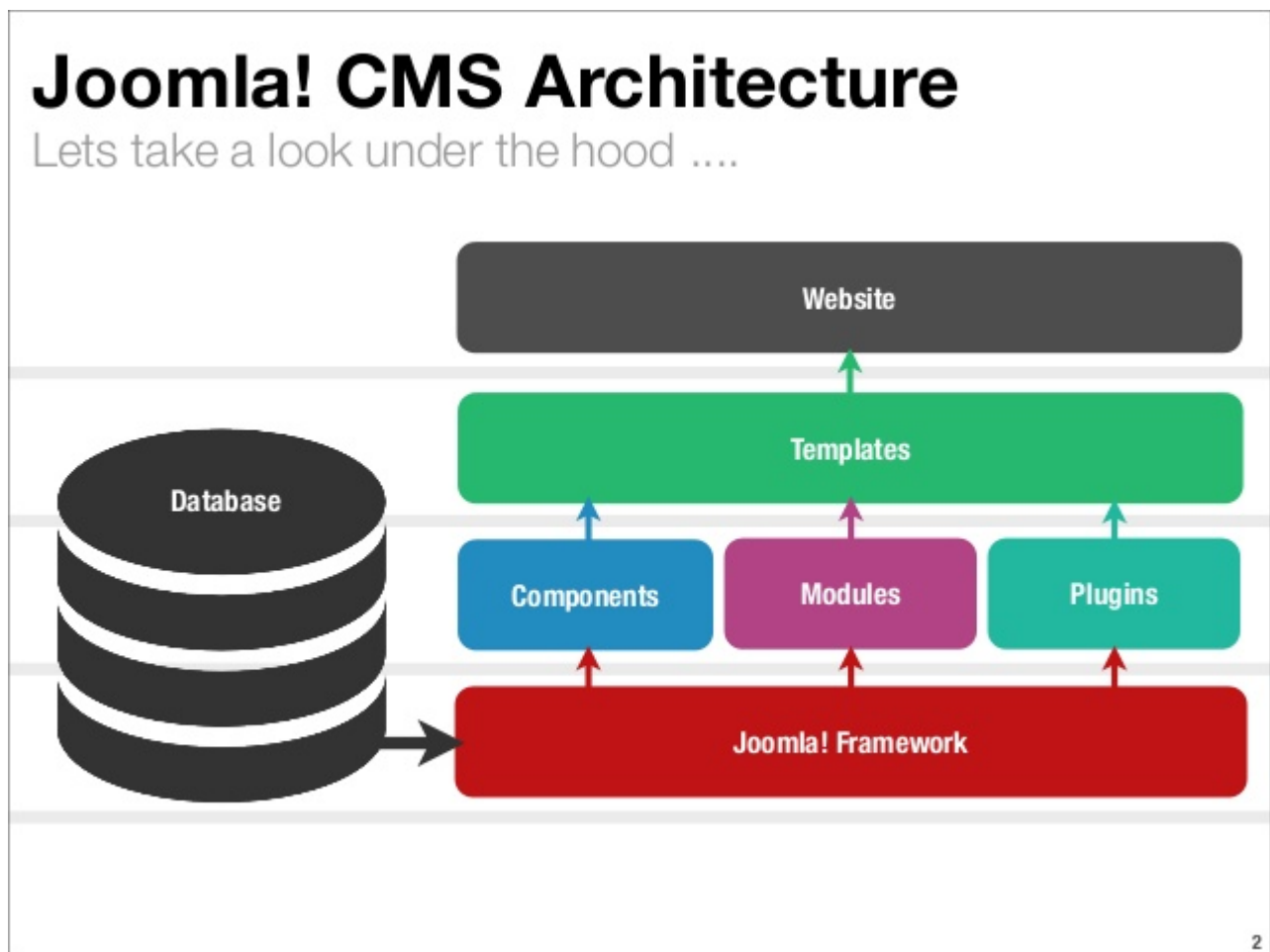
Frontend & Backend eines CMS

Ein CMS wird meist in ein Frontend- und ein Backend-Bereich unterteilt. Das Frontend eines CMS (wörtlich übersetzt „Vorder-Ansicht“) ist die Webseite, die ein Besucher angezeigt bekommt.



Das Backend eines CMS (wörtlich übersetzt „Hinter-Ansicht“) ist ein geschützter Bereich, in dem die Administration des CMS stattfindet. Hier stellt das CMS nach einer Anmeldung alle Funktionen zur Verfügung, die der jeweilige Benutzer zum Verwalten der Webseite benötigt. Häufig können für unterschiedliche CMS-Benutzer auch unterschiedliche Zugriffsrechte festgelegt werden, z.B. der Designer auf die Templates des CMS, der Redakteur auf die Inhalte des CMS.

CMS-Architektur anhand von Joomla



Datenbank

Alle Inhalte (Text, Menüs, Benutzer, ...) außer Bilddateien & Dokumente sind in einer Datenbank (z.B. MySQL) gespeichert.

Framework

Das Joomla-Framework (Rahmenstruktur/Programmiergerüst) ist ein modulares PHP-Framework und bietet eine Auswahl an vorgefertigten Funktionen und Bibliotheken, die einfach anzupassen bzw. leicht zu erweitern sind.

Extensions

Durch den modularen Aufbau des Joomla-Framework kann man dieses einfach mithilfe von Components, Modules, Plugins oder Templates erweitern.

Components

Komponenten sind die Anwendungen innerhalb von Joomla!. Unter einer Joomla! Komponente versteht man eigene Applikationen, die innerhalb von Joomla! laufen. Einige dieser Komponenten sind bereits in Joomla! integriert wie zum Beispiel das Artikelsystem zur Darstellung der Artikel oder eine einfache Weblinks-Verwaltung.

Komponenten sind also Anwendungen zur Verarbeitung und Darstellung von Informationen. Diese Komponenten unterliegen dabei einer eigenen Administration, in der Sie Ihre Daten verwalten oder auch die Verarbeitungsweise von vom Besucher eingegeben Informationen bestimmen.

Neben den bereits in Joomla! integrierten Komponenten findet man auf dem Markt eine Vielzahl kostenloser (das heißt lizenzfreier), aber auch einige kostenpflichtige Komponenten. Sehr bekannt sind Komponenten wie Foren, Gästebücher und Kommentarmöglichkeiten. Aber auch komplexe spezifische Anwendungen wie Kleinanzeigenmärkte, Buchungssysteme und Übersetzungstools für multilinguale Websites sind erhältlich.

Module

Unter Modulen bei Joomla! versteht man die Tools zur Anzeige von Informationen an verschiedenen Stellen auf der Seite. Diese Informationen bezieht Joomla! dabei in der Regel aus der Datenbank. Das heißt, sämtliche Anwendungen (Joomla! selbst, andere Komponenten), die ihre Daten in der Datenbank speichern, können abgefragt werden. Typische Beispiele sind das „wer ist online“-Modul, das die Anzahl der Besucher anzeigt oder das „latest news“-Modul, das die letzten Nachrichten kompakt in einer Liste anzeigt.

Module können aber auch dafür genutzt werden, selbst definierten Inhalt anzuzeigen: beispielsweise ein einfaches Bild oder einen eigenen Text.

Diese Module werden meist neben dem Hauptinhalt in einer Spalte angezeigt und lockern damit das Erscheinungsbild auf und liefern weitere Informationen.

Plugins

Plugins haben vielerlei Funktionen und arbeiten teilweise mit Komponenten zusammen. Der Editor ist beispielsweise ein Plugin, ebenso wie die Möglichkeit, Bilder einzubinden. Erweiterungen für alle drei Elemente, zum Beispiel umfangreichere Editoren oder erweiterte Sprachpakete, lassen sich zusätzlich installieren.

Templates

Unter Template versteht man nicht nur bei Joomla! den gestalterischen „Rahmen“ der Website.

Der Sinn eines Content Management Systems ist die Trennung von Inhalt und Design. In der Datenbank stehen nur die reinen Texte, das Template sorgt vor der Ausgabe auf dem Bildschirm dafür, dass diese - wie vom Designer festgelegt - „passend“ angezeigt werden.

Sinn und Zweck dieser Trennung liegt auf der Hand: Auch nicht mit der Web-Programmierung vertraute Nutzer können die Inhalte pflegen und müssen sich nicht um die Darstellung kümmern. Der Designer legt verschiedene Formatierungen fest, aus denen der Nutzer wählen kann (klein, groß, Überschrift, fett etc.), so dass diese auf der gesamten Website konsistent angezeigt werden.

Natürlich ist ein solches Template noch mehr: Es bestimmt den generellen Aufbau der Seite, ist also auch für den Gesamteindruck Ihrer Website verantwortlich. Denn die besten Inhalte nutzen nichts, wenn diese aufgrund einer nicht durchdringbaren Struktur unauffindbar bleiben. Ebenso sorgt das Template für den stimmigen Transport Ihrer CI.

Website

Zum Schluss entsteht eine Webseite, mit der die Benutzer & Besucher interagieren können.

Vorteile

Dezentralisierte Wartung

Ein gutes webbasiertes Content Management System funktioniert mit jedem gewöhnlichen Webbrowser (zum Beispiel Internet Explorer, Mozilla Firefox oder Opera). Sie benötigen also keine zusätzliche Software und können ohne Engpässe Ihre Internet-Präsentation warten und erweitern, wann und wo Sie wollen.

Trennung Inhalt & Gestaltung

Zum einen werden bei einem CMS die **Gestaltung der Webseite (Layout & Design)** und die **Verwaltung der Inhalte getrennt** verwaltet. Dadurch muss man sich beim Einsatz eines CMS nicht bei Inhalten um das Layout kümmern und umgekehrt. Außerdem kann so **mit wenigen Klicks** im CMS das **Layout der kompletten Webseite geändert** werden.

Für Autoren ohne technischen Background

Wer Textverarbeitung beherrscht, kann auch in einem modernen CMS System online Inhalte erstellen, Bilder veröffentlichen oder neue Produkte anlegen. Über ein modernes Content Management System können Sie ohne HTML-Kenntnisse Ihre Homepage pflegen.

Konfigurierbare Zugriffsbeschränkung

Ein CMS System beinhaltet in der Regel eine eigene Benutzerverwaltung. Benutzern können Rollen und Berechtigungen zugewiesen werden, wodurch die unautorisierte Veränderung von Inhalten effektiv verhindert wird.

Einhalten von Design-Vorgaben

In einem CMS können Inhalte vom Design getrennt gespeichert werden. Folglich wird der gesamte Content aller Autoren in einem Design zentral ausgegeben.

Automatische Navigations-Generierung

Menüs können in CMS Lösungen aus den Datenbankinhalten automatisch erzeugt bzw. generiert werden. Hyperlinks werden nur dargestellt, wenn sie auf gültige Seiten verweisen. Wenn beispielsweise diese Seiten gelöscht werden, können auch die Hyperlinks automatisch ausgeblendet werden.

Speicherung der Inhalte in einer Datenbank

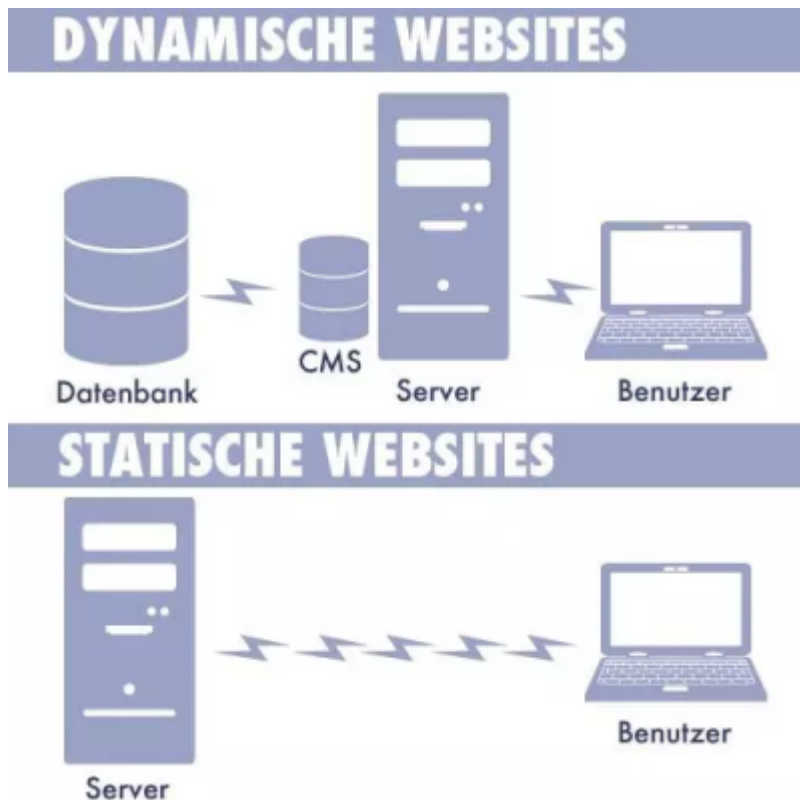
In einem guten CMS System werden die Inhalte in der zentraler Datenbank abgelegt. Eine zentrale Content-Speicherung bedeutet, dass Inhalte an verschiedenen Stellen wiederverwendet werden und für das jeweilige Medium (Webbrowser, Mobiltelefon/Wap, PDA, Print) adäquat formatiert werden können.

Dynamische Inhalte

Modulare Erweiterungen wie z.B. Foren, Umfragen, Shops, Anwendungen, Suchfunktionen und News-Management stehen als Module zum Einsatz bereit.

Content just in time

Die Publikation von Inhalten ist zeitgenau steuerbar. Inhalte können im Hintergrund vorbereitet und nur von berechtigten Usern in der sog. „hidden preview“ vorab eingesehen werden.



Nachteile

Eingeschränkte Gestaltung der Website

Durch die festgelegte Struktur eines CMS bestehen nicht vielen Freiräume was Widgets oder Features zur Integration in die Website betrifft. Es können nur verfügbare Features in die Website integriert werden.

Abhängigkeit vom Dienstleister

Entscheidet man sich für ein CMS, so ist man von der Aktualisierung und Service der Software vom jeweiligen Hersteller des Website-CMS abhängig.

Sicherheitsaspekt

Gerade **weitverbreitete Systeme (TYPO3, Joomla, Wordpress,...)** werden (genau wie bei den Betriebssystemen auch) natürlich **mit Vorliebe gehackt**. **Sicherheitslücken verbreiten sich schnell** in der Szene. Die vollständige Absicherung des Systems - zum Beispiel durch Ändern aller üblichen Installationspfade - kann sehr schnell sehr aufwändig werden. Allerdings sind bei den oben genannten Systemen auch die Entwickler sehr schnell mit der Erstellung von Patches. Es bedarf allerdings dann auch der entsprechenden Beobachtung der bekannten Foren und Mailinglisten.

Geringere Performance

Da die Inhalte dynamisch beim Aufruf der Seite generiert werden, kann der Auslieferung der kompletten Seite zumeist nur langsamer als bei einer statischen HTML-Seite erfolgen. Gute CMS-Systeme bieten hier allerdings geeignete Caching-Lösungen (Zwischenspeicher) an, die diesen Nachteil wieder ausgleichen.

Mehr Funktionalität als benötigt

Ein CMS muss als Gesamtsystem angesehen werden, in dem sich viele Programmteile bedingen oder voneinander abhängen. Will man das System um nicht benötigte Funktionen reduzieren kann es häufig zu unerwünschten Nebenwirkungen kommen.

Aufwendigere Backups

Genügt bei statischen Seiten meist ein einfaches Kopieren der Inhalte per FTP, so ist ein Backup eines CMS aufwändiger. Hier muss zusätzlich die Datenbank gesichert werden, sowie eventuell auch Systemeinstellungen des Servers. Auch das Rückspielen von Backups im Falle eines Datenverlustes sollte in regelmäßigen Abständen getestet werden.

Verbreitete CMS-Systeme

WordPress



Ursprünglich ein reines Blogsystem ist es heute mit über 70 Mio. Installationen das am weitesten verbreitete CMS.

Vorteile

- Installations- und Einrichtungsaufwand überschaubar
- Riesige Auswahl an kostenlosen bzw. -günstigen Designs
- Vielfältige Erweiterungsmöglichkeiten durch Plugins
- Ideal für SEO-Maßnahmen (sehr gute Tools: z.B. YOAST)

Nachteile

- WordPress braucht jede Menge Systemressourcen
 - Ladegeschwindigkeit bei hohem Traffic oft langsam
 - Viele Updates, teilweise mit Sicherheitsrisiken
-

TYP03



Content-Management-Systeme: Typo3-LogoDas „Open Source Flaggschiff“ mit mehr als 9 Mio. Installationen und 500.000+ Websites.

Vorteile

- Weitverbreitet, viele Experten und Entwickler
- Viele Funktionen
- Erheblich schneller als WordPress, auch bei hoher Besucherfrequenz

Nachteile

- Das Setup ist komplex und für den Laien ungeeignet
 - Auch bei Backend-Anpassungen sind profunde Kenntnisse im Administrationsbereich erforderlich
-

Joomla

Content-Management-Systeme: Joomla-LogoIst schon lange auf dem Markt und hat mit 1,2 Mio.+ Downloads eine eingeschworene Fangemeinde. Durch einen Entwicklungsstopp vor ein paar Jahren ist Joomla ein wenig ins Hintertreffen geraten. Derzeit holt dieses CMS allerdings wieder mächtig auf.



Vorteile

- Installation und Einrichtung einfach und gut dokumentiert
- Viele Erweiterungen und vorgefertigte Designs (ähnlich wie bei WordPress)

Nachteile

- Beliebtes Ziel von Hackern, nicht zuletzt aufgrund der vielen Erweiterungsmöglichkeiten (gilt übrigens auch für WordPress)
- Beim Rollen- und Rechtemanagement gibt's Luft nach oben

⇒ [2.01\) Joomla Installation](#)

Drupal

Content-Management-Systeme: Drupal-Logo Drupal ist ein Baukastensystem mit einer riesigen Auswahl an Features. Obwohl es unter die Content-Management-Systeme fällt, ist es eigentlich ein Content-Management-Framework (CMF). Drupal funktioniert im Grunde wie Lego – man kann alles damit bauen.



Vorteile

- Drupal verfügt über ein differenziertes Rollen- und Rechtesystem
- Es hat jede Menge Funktionen, die als Baustein in das System integriert werden können
- Das Backend ist voll individualisierbar

Nachteile

- Das Drupal-Setup ist vergleichsweise kompliziert
 - Eingriffe im Backend und am Server erfordern Routine
-

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:17



Last update: **2019/05/06 19:20**

↓
Kapitel NETZWERKE als PDF exportieren

NETZWERKE

Allgemeine Netzwerkgrundlagen

- Grundlagen
- Topologien
- Übertragungsmedien
- Ethernet + Zugriffsverfahren CSMA/CD
- Schichtenmodell
- Netzwerkgeräte
- Adressierung
 - Adressierung Übungen
- Routing
- Netzwerkbefehle
- Protokolle

Übungsaufgaben (Praxis)

- Aufbau von Netzwerken
- Routing
- Netzwerkübungen (Subnetting, DNS-Server, DHCP-Server, WEB-Server, Mail-Server)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:18

Last update: **2019/05/04 10:36**



Matura-Umfeldfragen

Umfeldfragen

1. Erläutere den Begriff „Cyber-Mobbing“

Absichtliches und über einen längeren Zeitraum anhaltendes Beleidigen, Bedrohen, Bloßstellen, Belästigen oder Ausgrenzen anderer über digitale Medien.

Besonderheiten: Jederzeit, großes Publikum, Anonymität

1. Erkläre verschiedene Möglichkeiten, sich gegen Cyber-Mobbing zu wehren.
 - [Cyber-Mobbing Reaktion](#)
2. Nenne verschiedene Anlaufstellen zur Unterstützung und Beratung bei Cyber-Mobbing.
 - [Beratungsmöglichkeiten](#)
3. Erläutere den Begriff „Datenschutz“.
 - [Datenschutz](#)
4. Nenne verschiedene Grundregeln, persönliche Daten im Internet zu schützen.
 - [Grundregeln Daten schützen](#)
5. Erläutere den Terminus „Recht am eigenen Bild“.
 - [Recht am eigenen Bild](#)
6. Gib an, wie man am besten vorgehen sollte, um peinliche Fotos wieder aus dem Netz zu bekommen.
 - [Fotos entfernen](#)
7. Nenne Grundregeln zur Erstellung sicherer Passwörter.
 - [Sichere Passwörter](#)
8. Erläutere den Begriff „Zwei-Stufen-Authentifizierung“.
 - [Zwei-Stufen-Authentifizierung](#)
9. Erläutere, was durch das Urheberrecht geschützt wird und wann Urheberrechte verletzt werden.
 - [Urheberrecht](#)
10. Erläutere den Begriff „Creative-Commons-Lizenz“.
 - [Creative-Commons-Lizenz](#)
11. Erläutere den Begriff „Phishing“ und wie man sich davor schützen kann.
 - [Phishing](#)
12. Erläutere den Begriff „Sextortion“.
 - [Sextortion](#)
13. Erläutere den Begriff „Cyber-Grooming“.
 - [Cyber-Grooming](#)
14. Nenne verschiedene Betrugsarten, auf die achtlose Internetuser reinfallen können.
 - [Betrugsarten](#)
15. Nenne verschiedene Möglichkeiten, seinen Computer vor Viren, Spam, etc. zu schützen.
 - [Viren, Spam](#)
16. Erläutere den Begriff „Spam“ und wie man sich davor schützen kann.
 - [Spam](#)
17. Erläutere den Begriff „Hoax“.
 - [Hoax](#)
18. Nenne verschiedene Möglichkeiten, sicher im Internet einzukaufen/zu bezahlen.

- [Bezahlen im INternet](#)

19. Erläutere, inwieweit Kinder bzw. Minderjährige im Internet einkaufen dürfen.

- [Altersgrenzen Einkaufen im Internet](#)

20. Erläutere den Terminus „Recht auf Gewährleistung“.

- [Recht auf Gewährleistung](#)

From:

<http://elearn.bgamstetten.ac.at/wiki/> - **Wiki**

Permanent link:

http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201819_matura:19



Last update: **2019/05/14 15:14**