Technische Informatik - Rechnerarchitektur - Teil 1

Wir erinnern uns an Klasse 10 - Teilgebiete der Informatik:

Theoretische Informatik:	Algorithmusbegriff, Sprachen, Grammatiken, Automaten, Grenzen, Komplexität usw.
Praktische Informatik:	Algorithmenerstellung, Programmierung, Übersetzerbau (Compiler, Interpreter), Betriebssysteme usw.
Technische Informatik:	Rechneraufbau- und funktion, Aufbau und Entwicklung von Computerkomponenten, Entwurf von Schaltungen, Computernetze usw.
Angewandte Informatik:	Anwendung von Methoden und Ergebnissen der anderen 3 Bereiche (*Kerninformatik*) auf Probleme in allen Gesellschaftsbereichen, Nutzung von Anwendungssystemen zur Lösung von Aufgaben (*Informatiksysteme*)

Alles das, was grün geschrieben ist, haben wir schon mehr oder weniger tiefgehend behandelt, so dass ihr einen Einblick und Überblick über die Informatik habt.

Angewandte Informatik besteht dabei in der Nutzung von Netzwerken, Office-Systemen, dem TINspire in Mathe und Rasenmäh-Robotern im Garten (... *jetzt wo ihr viel Zeit habt, helft ihr euren Eltern und Großeltern hoffentlich bei der Gartenarbeit* ...). Na ja, eigentlich besteht der Informatikanteil eher in der Bereitstellung dieser Dinge ... nicht dass wir noch eure heiß geliebte Erstellung von Vorträgen mit Powerpoint als Beitrag zum Informatiklernen bezeichnen.

In der "Technischen Informatik " haben wir allerdings etwas Nachholebedarf, da wir bisher nur grob über die Funktionsweise von Computern gesprochen haben (in Klasse 7 und 10).

Was soll in dieser Thematik unser "Fahrplan" sein (ggf. coronaabhängig änderbar)?

- Wiederholungen zum groben Computeraufbau und zu Computerkomponenten
- Wiederholungen und Übungen zur Codierung (natürliche Zahlen, Text, Bilder, Sound)
- Codierung von "Ganzen Zahlen" (Zweierkomplement) und "Reellen Zahlen" (Gleitpunktcodierung), Übungen dazu
- Kennenlernen des "Von-Neumann-Modells", welches den Aufbau und die Funktionsweise von Computern beschreibt, moderne Abweichungen vom Modell
- Kennenlernen der logischen Funktionsweise eines Von-Neumann-Rechners (also eines vereinfachten Computers) anhand der Assemblerprogrammierung (vereinfacht), Entwurf und Umsetzung einfacher Assemblerprogramme

- Grundlagen der E-Lehre zum Entwurf von Schaltungen (U, I, R, Schaltungsarten)
- Leiter und Halbleiter, Dotierung von Halbleitern, Dioden, bipolare Transistoren und MOSFET-Transistoren
- Entwicklung und Realisierung logischer Grundschaltungen (Gatter) aus Transistoren, "Boolesche Funktionen" zur Beschreibung der Schaltfunktion
- Entwurf von Schaltnetzen mittels Boolescher Funktionen, algebraischer Umformungen und Gattern, Realisierung mit Crocodile Physics (Schaltungssimulation)
- Schaltwerke (Speicherschaltungen) aus Gattern realisieren und verstehen, Kombination mit Schaltnetzen zur Erstellung einer Addiererschaltung

Wiederholungen zum groben Computeraufbau und zu Computerkomponenten:

Aufgabe: Wiederhole den grundsätzlichen Aufbau und die grundsätzlichen Komponenten eines Computers! Ergänze Bezeichnungen, Aufgaben, Zusammenhänge! Nutze den Hefter aus Klasse 10, Bücher, Internet!

Mainboard:

Schnittstellen:

Netzteil:

CPU:

RAM:

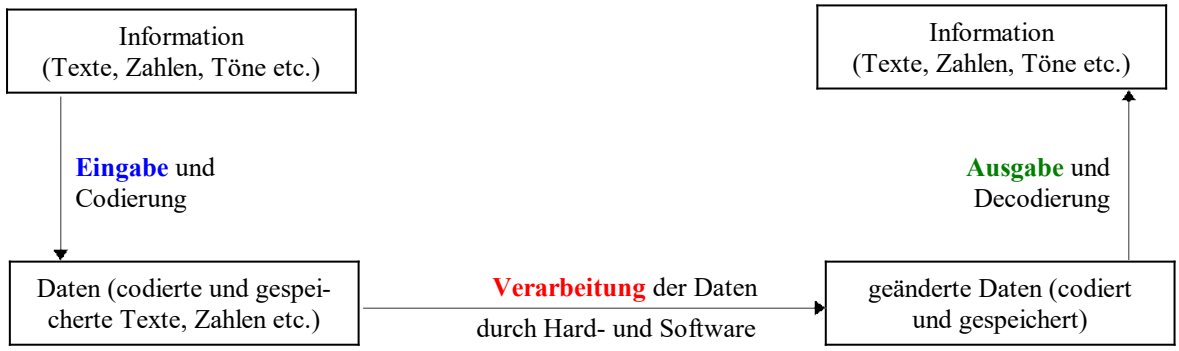
HD/SSD:

Slots/Steckkarten:

Bussystem:

Wiederholungen und Übungen zur Codierung:

Erinnern wir uns an das E-V-A-Prinzip zur Datenverarbeitung:



Dabei spielen die Codierung und Decodierung eine wesentliche Rolle, da wir Menschen am besten mit analogen Informationen umgehen können (Bilder, Farben, Töne, ...), Computer jedoch auf Grund ihrer technischen Struktur (Transistoren als Ein-/Ausschalter) nur mit digitalen Daten (1/0).
Deswegen müssen sämtliche Informationen, die mit einem Computer verarbeitet werden sollen in das Binärsystem (Dualsystem) codiert und dann gespeichert werden.

Dazu haben wir in Klasse 10 folgendes gelernt (Hefter benutzen!):

Grafik: Im Pixelformat (z.B. bmp) wird die Farbe jedes Bildpunktes (Pixel) eines Bildes als Zahl codiert (mit je 16 bit, 32 bit o.ä.).
Im Vektorformat (z.B. svg) besteht ein Bild aus verschiedenen Objekten, von denen die verschiedenen Eigenschaften (Position, Farbe, Linienart usw.) als Zahlen codiert werden.

Sound: Ein Sound (eine Wellenform) wird durch Sampling in einzelne Messpunkte zerlegt, deren Messwerte als Zahlen codiert werden.
Beispiel: CD-Aufnahmen (44100 Messpunkte pro Sekunde, je Messpunkt Speicherung des Wertes mit 16 bit)

Texte: Textzeichen werden über Codetabellen (z.B. ASCII, ANSI als Erweiterung von ASCII, Unicode und deren häufigste UTF-8- bzw. UTF-16-Codierung) Zahlenwerten zugeordnet, die codiert werden.

Zahlen: Natürliche Zahlen werden in das Binärsystem (Dualsystem) codiert. Dieses ist wie auch unser Dezimalsystem ein Stellenwertsystem (Positionssystem).

Als Erinnerung und falls ihr es nicht mehr besitzt, findet ihr ein Lernblatt dazu als zweiten Download auf der Internetseite.

Ihr schaut euch nun die damals vermittelten Inhalte wiederholend an und löst dann folgende Aufgaben dazu schriftlich und nachvollziehbar.

Hinweis: 1 KB=2^{10} Byte, 1 MB=2^{20} Byte, 1 GB=2^{30} Byte, 1 TB=2^{40} Byte

---

A1: Codiere das Wort „Abba“ im erweiterten 8-Bit-ASCII-Code!

A2: Ein Bild der Größe 1600 Pixel x 900 Pixel ist mit einer Farbtiefe von 16 bit gespeichert. Wie viel verschiedene Farben kann das Bild theoretisch enthalten?
Berechne den Mindestspeicherbedarf des Bildes und gib ihn in Bit, Byte, KB und MB an! Warum benötigt die Bilddatei auf einer Festplatte jedoch etwas mehr als den berechneten Speicherplatz?

A3: Die theoretisch maximale Speicherkapazität einer CD-ROM beträgt 879 MB. Wieviel Minuten Musik im CD-Format (Abtastfrequenz 44100 Hz, 16 bit -Auflösung, stereo, unkomprimiert) lassen sich auf einer solchen CD unterbringen?

A4: Codiere die folgenden Zahlen als 16-bit-Binärzahl!

$8761_{(10)}$ // $3AF7_{(16)}$

A5: Decodiere die folgenden Zahlen als Dezimalzahlen!

$1101\ 1011_{(2)}$ // $2E0A_{(16)}$

A6: Gegeben sei die nebenstehende Pixelgrafik. Ermittle den minimalen unkomprimierten Speicherbedarf dieser Grafik!
Beschreibe anhand der Grafik eine Möglichkeit, Speicherplatz durch Komprimierung einzusparen!