

Pro Scientia Referat: „Der beste Spiegel für den eigenen Gedankenprozess“

Der Spiegel und meine Interpretation:

Dem ausgefallenen Titel geschuldet bleibt ein großer Spielraum für Interpretation, was dieser „Spiegel“ denn ist. Geprägt von meinen beruflichen Problemstellungen, akademischen Fokus und Freizeitaktivitäten ist es für mich ein Handwerk: Softwareprogrammierung – Also das „Verfassen“ von Computerprogrammen. Dabei stellt ein Computerprogramm eine Folge an Anweisungen dar, um bestimmte Funktionen beziehungsweise Aufgaben oder Probleme mithilfe eines Computers zu bearbeiten (vgl. ISO/IEC 2382-1:2015 [1]). Was diese Norm nicht beschreibt, ist wie die Anweisungen auszusehen haben. Somit unterliegt zwar die Sprache des Programmierens gewissen Regeln, die Problemlösung ist aber höchst individuell und hängt vor allem davon ab, wie ich das Problem gedanklich löse und dann für den Computer niederschreibe. Diese „Niederschrift“ ist dann mein eigener Gedankenprozess, den ich am Bildschirm vor mir aktiv betrachten kann. Für mich persönlich bietet genau das die Möglichkeit, meinen Gedankenprozess nachzuvollziehen, oder auch zu verbessern. Damit wir aber überhaupt unsere Gedanken in einer bestimmten Syntax an einen Computer weitergeben können, braucht es entsprechende Computer Hard- und Software, welcher sich diese Zusammenfassung im folgenden Teil widmet.

Softwareprogrammierung wird heute meist mittels „höheren“ Programmiersprachen [2] umgesetzt. Eine höhere Programmiersprache zeichnet sich vor allem durch die Verwendung von für den Programmierer einfachen Wortbefehlen, die aus der englischen Sprache entnommen sind, aus und erleichtert somit die Softwareprogrammierung. Leistungsstarke Hardware ermöglicht die Umsetzung von relativ anspruchsvollen Anwendungen und Problemlösungen. Die Meilensteine, die für diesen „Ist-Stand“ meiner Meinung nach notwendig waren, sind im folgende aufgelistet. Diese persönliche Auflistung erhebt natürlich keinen Anspruch auf Vollständigkeit.

Die Anfänge, mechanische Hardware und der erste „allgemeine Ansatz“:

Den ersten Grundstein legt der „Abakus“ – Erste Bauarten wurden in etwa 2000 v.Chr. eingeführt – Die heute bekannte Form als „Rechenschieber“, wie in der nebenstehenden Abbildung gezeigt, wird in etwa ab 1300 v.Chr. eingeführt. Der Abakus erleichtert das Berechnen von großen, für den Menschen schwer merkbaren Zahlen und kann somit als erste Computer-Hardware angesehen werden. 1642 n.Chr. wird mittels einfacher Mechanik von Blaise Pascal die „Pascaline“ entwickelt, die das Summieren von Zahlen ermöglicht. Das Konzept wird von Gottfried Leibniz weiterentwickelt und ebenfalls im gleichen Zeitraum als „Leibnizrechenmaschine“ vorgestellt, welche eine verlässlichere Mechanik vorweist und

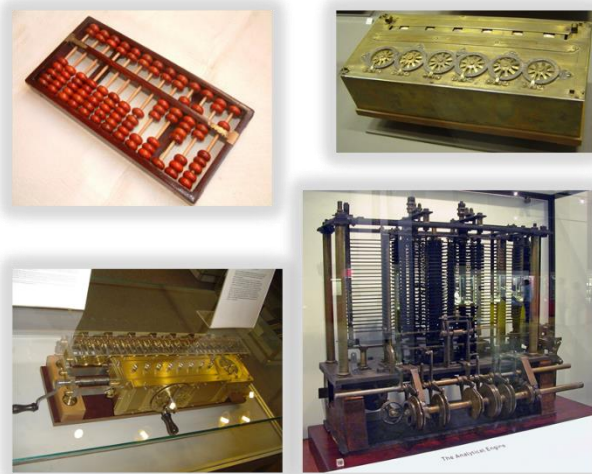


Abbildung Fehler! Kein Text mit angegebener Formatvorlage im Dokument...1: Chinesischer Abakus (oben links), Pascaline (oben rechts), Leibniz Rechenmaschine (unten links) und "Analytical Engine" (unten rechts)

auch das Multiplizieren ermöglicht. Den ersten Versuch eines Computers, der beliebige Berechnungen ausführen kann und somit allgemein einsetzbar ist, stellt Charles Babbage „Analytical Engine“ dar - Eine Maschine, die beliebige Berechnungen mittels Lochkartenanweisung ermöglichen soll. Sie wurde aber nie gebaut.

Soft- und Hardwareentwicklung im 20. Jahrhundert:

Der erste, wirklich gebaute Rechner für beliebige Berechnungen war der Electronic Numerical Integrator and Computer (ENIAC), der 1946 vom US Militär für die Flugbahnberechnung von Artilleriegeschossen verwendet wurde. Soft- und Hardwaretechnisch ist dieser Computer aber noch bei weitem nicht auf heutigem Niveau, da in der ENIAC Vakuumröhren als Hardware dienten und auch die Programmierung noch rudimentär mit Lochkarten beziehungsweise (in der Ursprungsform der ENIAC) noch über das individuelle Verdrahten der Vakuumröhren funktionierte [3].

Lange Zeit konnte man Computerhardware aufgrund der langen Schaltzeiten von mechanischen Schaltern nicht essentiell verbessern. Die Schaltzeit (in diesem Falle definiert als die Zeit, die zwischen dem Ein- und Ausschalten, oder anders formuliert zwischen dem Wechsels von Zustand „null“ auf Zustand „eins“ vergeht) von mechanischen Schaltern war zu lange, um komplexere Berechnungen „zeitnah“ zu ermöglichen. Erst die Einführung von Transistoren und später integrierten Schaltkreisen ermöglicht Computerhardware mit den heutigen Geschwindigkeiten. In Tabelle 1 sind die Schaltzeiten für die erwähnten Bauteile nochmals aufgelistet. Der erste Computer, der durch die Verwendung von Transistoren extreme Vorteile im Hinblick auf Rechengeschwindigkeit und Ausfallsicherheit gegenüber der ENIAC hatte, war der IBM 7090. Mit 1976 finden auch erstmals Vektorprozessoren, die heute vor allem für wissenschaftliche Berechnungen verwendet werden, im weltweit ersten Supercomputer, dem „Cray-1“, Anwendung.

Mit der zunehmenden Leistungsfähigkeit der Hardware, stieg die Anzahl an bearbeitbaren Problemen, die auch immer komplexer wurden. Die bisherige Befehlsanweisung über Lochkarten (und früher auch noch Binärcode) machte das Verfassen von Computerprogrammen äußerst umständlich und fehleranfällig. 1957 wurde bei IBM die erste „höhere Programmiersprache“ entwickelt. Formula Translator (kurz: FORTRAN) erlaubte die Verwendung von einfachen Begriffen wie „print“ oder „goto“, Entscheidungen mittels „if/then/else“, welche die Softwareprogrammierung erleichterten. In den späten 80er Jahren wurde mit C++ die erste „objektorientierte“ Sprache entwickelt. Dieses Konzept der Softwareprogrammierung findet auch heute noch in aktuellen Programmiersprachen wie Python, C#, Matlab und vielen weiteren Anwendung.

Bauteil	Schaltzeiten
Relais	0.1 s
Vakuumröhre	0.0001 s
Transistor	0.000001 s
Integrierter Schaltkreis	0.000000001 s

Tabelle 1: Schaltzeiten für mechanische und elektrische Bauteile

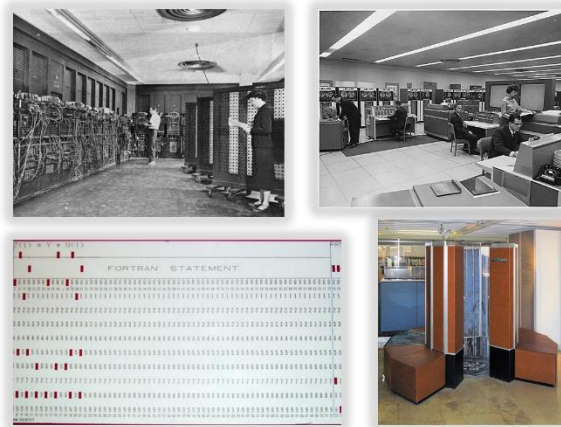


Abbildung Fehler! Kein Text mit angegebener Formatvorlage im Dokument.:2: ENIAC (oben links), IBM 7090 (oben rechts), „Hello World“ als Lochkarte für FORTRAN (unten links) und Cray 1 (unten rechts)

Leistbare, kleine Hardware:

2012 wird mit der Einführung des Raspberry Pi, einem Einplatinencomputer für umgerechnet € 30, die Hard- und Softwareentwicklung des 20. Jahrhunderts für die breite Masse zugänglich gemacht. Der niedrige Preis ermöglicht die Anwendung des Raspberry Pi in Bildungseinrichtungen und bietet somit früh den Zugang zu Softwareprogrammierung. Die Verkleinerung der Computer bildet auch den Anfang von wichtigen Trends des 21. Jahrhunderts, wie Industry 4.0, Internet of Things, sowie Machine Learning und Predictive Maintenance [4]



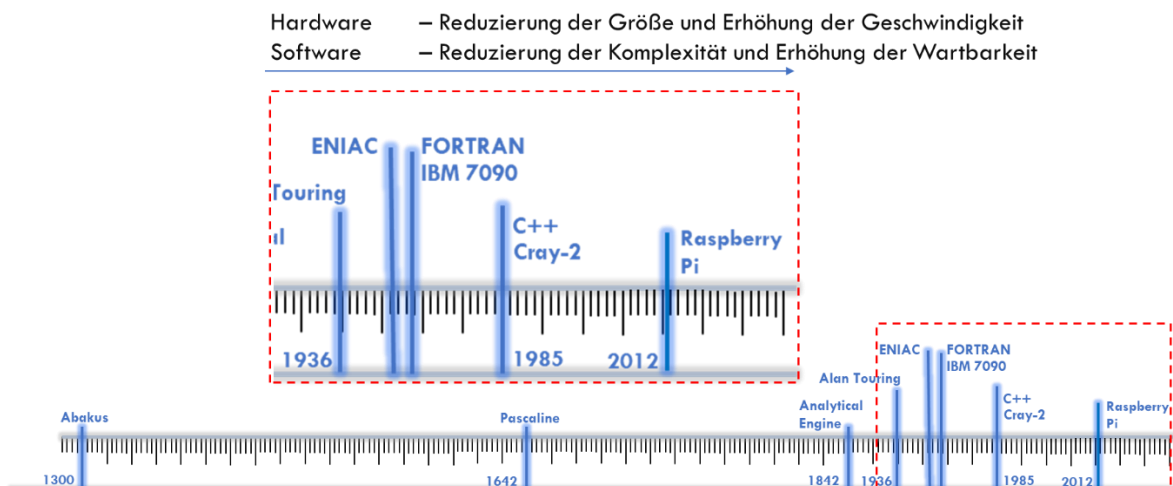
Abbildung Fehler! Kein Text mit angegebener Formatvorlage im Dokument.:3: Raspberry Pi (Model 2)

Conclusio:

Abschließend lassen sich, wenn man die oben erwähnten Meilensteine auf einer Zeitlinie aufträgt, zwei wichtige Schlüsse ziehen. Erst die Einführung von elektrischen Schaltbauteilen ermöglicht einen exponentiellen Zuwachs von Rechenleistung. Somit findet sich, ausgenommen von Abakus, Pascaline und Leibnizrechenmaschine, die wichtigsten Meilensteine der Hardware in der zweiten Hälfte des 20. Jahrhunderts wider – Nach der Einführung des Transistors.

Für die Software lässt sich vor allem die Reduzierung von Codeanweisungen durch die Einführung von höheren Programmiersprachen und objektorientierter Programmierung herauslesen. Beides dient dem Zweck, dass die Komplexität im Code reduziert und die Wartbarkeit erhöht wird.

Erst dieses Zusammenspiel aus Hard- und Softwareentwicklung vor allem im 20. und 21. Jahrhundert ermöglicht die heutige, einfache Anwendung von Softwareprogrammierung, die jedem im Bezug auf die Analyse des eigenen Problemlösungsverhaltens wärmstens ans Herz gelegt sei.



Literaturverzeichnis:

- [1] ISO/IEC 2382-1:2015. ISO/IEC JTC 1 - Information technology
- [2] Clean Code: A Handbook of Agile Software Craftsmanship. Robert C. Martin, Prentice Hall 2008.
- [3] Alan Turing: The Enigma. Andrew Hodges, Princeton University Press, 2014.
- [4] Thank You for Being Late: An Optimist's Guide to Thriving in the Age of Accelerations. Thomas L. Friedman, Farrar Strauss & Giroux 2016.

Fehler! Verweisquelle konnte nicht gefunden werden.