

# **ICT VIRTUALISIERUNG**

---

**Johann Widmer**

## ICT Virtualisierung

1. [Table of Contents](#)
2. [Titel](#)
3. [Der Autor](#)
4. [Vorwort](#)
5. [1. Rechnerarchitekturen](#)
6. [2. Virtualisierungsmodelle und Technologien](#)
7. [2.1 Was ist Virtualisierung?](#)
8. [2.2 Virtualisierungsansätze \(Übersicht und Definitionen\)](#)
9. [2.3 Abgrenzung der Virtualisierung](#)
10. [2.4 Technologie der Virtualisierung](#)
11. [2.5 Servervirtualisierung](#)
12. [2.6 Betriebliche Strategien, Prozesse, Anforderungen](#)
13. [2.7 Vorteile und Nachteile von Virtualisierungen](#)
14. [2.8 Grenzen der Virtualisierung](#)
15. [2.9 Übungen](#)
16. [2.10 Cases](#)
17. [3. Wirtschaftlichkeit und Ressourceneinsatz](#)
18. [3.1 Vorbemerkung zur Wirtschaftlichkeit](#)
19. [3.2 Auswirkungen der Virtualisierung auf die Wirtschaftlichkeit](#)
20. [3.3 Energieverbrauch / Energieeffizienz](#)
21. [3.4 Kennzahlen / Beurteilung der Virtualisierungseffizienz](#)
22. [3.5 Case Study](#)
23. [4. Technische Virtualisierungskonzepte](#)
24. [4.1 Was ist ein Konzept?](#)
25. [4.2 Case: Eine Systemintegration unter Berücksichtigung der Energieeffizienz](#)
26. [5. Technische und organisatorische Abläufe bei Virtualisierungslösungen](#)

27. [5.1 Organisatorische Abläufe in virtualisierten Umgebungen](#)
28. [5.2 Beurteilen von virtuellen Lösungen bezüglich Infrastruktur](#)
29. [5.3 Beurteilen von virtuellen Lösungen bezüglich Sicherheit](#)
30. [5.4 Cases](#)
31. [6. Praxisteil, Semesterauftrag](#)
32. [6.1 Vorbemerkung und Rahmenbedingungen](#)
33. [6.2 Auftrag](#)
34. [6.3 Disposition](#)
35. [6.4 Das Beurteilungsraster / Literaturhinweise](#)
36. [Bilderverzeichnis](#)

# Table of Contents

[Titel](#)

[Der Autor](#)

[Vorwort](#)

[1. Rechnerarchitekturen](#)

[2. Virtualisierungsmodelle und Technologien](#)

[2.1 Was ist Virtualisierung?](#)

[2.2 Virtualisierungsansätze \(Übersicht und Definitionen\)](#)

[2.3 Abgrenzung der Virtualisierung](#)

[2.4 Technologie der Virtualisierung](#)

[2.5 Servervirtualisierung](#)

[2.6 Betriebliche Strategien, Prozesse, Anforderungen](#)

[2.7 Vorteile und Nachteile von Virtualisierungen](#)

[2.8 Grenzen der Virtualisierung](#)

[2.9 Übungen](#)

[2.10 Cases](#)

[3. Wirtschaftlichkeit und Ressourceneinsatz](#)

[3.1 Vorbemerkung zur Wirtschaftlichkeit](#)

3.2 Auswirkungen der Virtualisierung auf die Wirtschaftlichkeit

3.3 Energieverbrauch / Energieeffizienz

3.4 Kennzahlen / Beurteilung der Virtualisierungseffizienz

3.5 Case Study

4. Technische Virtualisierungskonzepte

4.1 Was ist ein Konzept?

4.2 Case: Eine Systemintegration unter Berücksichtigung der Energieeffizienz

5. Technische und organisatorische Abläufe bei Virtualisierungslösungen

5.1 Organisatorische Abläufe in virtualisierten Umgebungen

5.2 Beurteilen von virtuellen Lösungen bezüglich Infrastruktur

5.3 Beurteilen von virtuellen Lösungen bezüglich Sicherheit

5.4 Cases

6. Praxisteil, Semesterauftrag

6.1 Vorbemerkung und Rahmenbedingungen

6.2 Auftrag

6.3 Disposition

6.4 Das Beurteilungsraster / Literaturliste

Bilderverzeichnis

# Titel

Johann Widmer  
ICT-Virtualisierung

Eine Einführung in die Virtualisierung von Informatik-Infrastrukturen

3. Auflage 2021

Johann Widmer ICT-Virtualisierung

Eine Einführung in die Virtualisierung von Informatik-Infrastrukturen  
ISBN - siehe Umschlag

Umschlag: J. Widmer

Grafiken und Satz: Johann Widmer, Zürich 3. Auflage 2021 (2020, 2016)

Copyright © 2016 Text, Illustration, Ausstattung by Johann Widmer

Alle Rechte vorbehalten. Das Werk und seine Teile sind urheberrechtlich geschützt. Jede Verwertung in anderen als den gesetzlich zugelassenen Fällen bedarf deshalb der vorherigen schriftlichen Einwilligung des Verfassers.

Nicht in allen Fällen war es möglich, die Rechteinhaber der Abbildungen und Texte ausfindig zu machen. Berechtigte Ansprüche werden selbstverständlich im Rahmen der üblichen Vereinbarungen abgegolten.

# Der Autor

Johann Widmer

Der Autor besitzt eine langjährige Industrieerfahrung in Telematik und Verfahrenstechnik, Audits und Controlling, Management Coaching, strategisch-tech- nologische Beratung, Geschäftsentwicklung, Forschung und Entwicklung, Workshop- Leitung und Training.

Er ist auf verschiedenen Stufen Dozent für Telematik, virtualisierte ICT-Architekturen, Architektur von Informations- und Kommunikationssystemen, für Netz- werk-Design, Betriebssysteme und Informations-Sicherheit.

Autor verschiedener Bücher und wissenschaftliche Veröffentlichungen in den Bereichen Telematik, mechanische Energiespeicherung und Materialwissenschaften.

# Vorwort

„Virtualität ist die Eigenschaft einer Sache, nicht in der Form zu existieren, in der sie zu existieren scheint, aber in ihrem Wesen oder ihrer Wirkung einer in dieser Form existierenden Sache zu gleichen. Das Wort führt über den französischen Begriff virtuell (fähig zu wirken, möglich) zurück auf das lateinische Wort virtus (Tugend, Tapferkeit, Tüchtigkeit, Kraft, Männlichkeit). Virtualität spezifiziert also eine gedachte oder über ihre Eigenschaften konkretisierte Entität, die zwar nicht physisch, aber doch in ihrer Funktionalität oder Wirkung vorhanden ist.“[1](#)

Dieses Lehrbuch basiert einerseits auf [1], [2] für die Höheren Fachschulen der Schweiz und andererseits auf der Vorlesung „Virtuelle Systeme“ für Master of Advanced Studies (MAS) der Fachhochschulen der Schweiz. In einem ersten Kapitel werden zuerst die grundlegenden Konzepte der heutigen Rechnerarchitekturen (Hardware) mit deren Betriebssoftware (Betriebssysteme und Applikationen).

In einem zweiten Kapitel befassen wir uns mit den Virtualisierungsmodellen. Die Studierenden werden in die Lage versetzt, die unterschiedlichen Modelle der Virtualisierung zu erklären sowie Vor- und Nachteile für eine zu wählende Lösung im Unternehmen zu bestimmen und die verschiedenen Arten, auf welche die einzelnen Elemente virtualisiert werden können zu erläutern.



Die Auswirkungen der verschiedenen Virtualisierungsansätze auf die betrieblichen Prozesse zu benennen und die betrieblichen Anforderungen an eine neue Umgebung und bestehende Vorgaben zusammen zu stellen und wissen, wie daraus ein Kriterienkatalog zu erarbeiten ist. abgestimmt auf die Zielsetzungen eines Unternehmens soll die Eignung von Virtualisierungslösungen für spezifische Einsatzbereiche im Unternehmen (Anwendungen, Desktop, Server) beurteilt werden und daraus eine Empfehlung abgeleitet werden, welche Lösung realisiert werden soll. Dies ergänzt die Ausführungen in diesem Kapitel.

Die in Frage kommenden Virtualisierungsansätze mit dem Geschäftsumfeld, den bestehenden Rahmenbedingungen und den betrieblichen Vorgaben sind in Zusammenhang zu bringen und die Konsequenzen zu bewerten.

Ein drittes Kapitel widmen wir der Wirtschaftlichkeit der Virtualisierung. Die Studierenden werden in die Lage versetzt, die Auswirkungen der Virtualisierung in Bezug auf Infrastruktur und Ressourcen wie Hardware, Lizenzen, Raumbedarf sowie Personal zu benennen.

Die Auswirkungen unterschiedlicher Virtualisierungsmodelle in Bezug auf die schonende Nutzung nicht erneuerbarer Ressourcen, den Stromverbrauch der benötigten Komponenten sowie mögliche sekundäre Energieeinsparungen und den Begriff der „Energieeffizienz“ zu erklären.

Aussagekräftige Kennzahlen zur Beurteilung der Virtualisierungseffizienz zu ermitteln und Massnahmen zur Erfassung der Verbrauchsdaten und Kennzahlen festzulegen.

Ein viertes Kapitel befasst sich mit den technischen Virtualisierungskonzepten. Die Studierenden werden in die Lage versetzt, die Grundlagen der Systemintegration zu verstehen und zu wissen wie beim Erstellen eines Konzeptes alle betroffenen Bereiche mit einzubeziehen sind.

Es werden Vorgaben zur Umsetzung energieeffizienter Lösungen benannt und die Studierenden wissen, wie diese konkret umgesetzt werden können.

Ein fünftes Kapitel führt technische und organisatorische Abläufe und Betriebskonzept auf. Die Studierenden werden in die Lage versetzt, die Anforderungen einer Virtualisierungslösung an die Infrastruktur, wie Raumbedarf, Kühlung, Anschlüsse, Energiebedarf etc. zu beschreiben und die Auswirkungen der Virtualisierung auf bestehende, betriebliche Prozesse zu benennen und zu wissen, wie diese anzupassen sind. Ein technisches und ein Betriebskonzept wird erstellt.

In einem sechsten Kapitel wird eine praktische Umsetzung einer Virtualisierungslösung als Semesterauftrag ausgeschrieben. Vertieft werden dabei die folgenden Inhalte:

- Die Hard- und Software Voraussetzungen bestimmen für Servervirtualisierung, Desktopvirtualisierung, Applikationsvirtualisierung, Speichervirtualisierung und Netzvirtualisierung.
- Ein Konzept zur Integration der gewählten Virtualisierungslösung umzusetzen
- Betriebliche Vorgaben, die sicherstellen, dass Sicherheitsmassnahmen eingehalten werden. Festgelegte Massnahmen implementieren

- Verwaltungswerkzeuge, um die Virtualisierungslösung zu betreiben einzusetzen
- Installieren und konfigurieren einer Virtualisierungslösung basierend auf den Vorgaben des technischen Konzepts sowie des Betriebskonzeptes.
- Basierend auf der Virtualisierungslösung geforderte Dienste über ein Netzwerk bzw. in einer Cloud bedarfsgerecht bereitstellen.

---

<sup>1</sup> Aus Wikipedia: <https://de.wikipedia.org/wiki/Virtualität>

# 1. Rechnerarchitekturen

Um die Virtualisierung von Computerhardware, von Netzen, von Speichern, von Betriebssystemen und von Applikationen besser zu verstehen, machen wir zuerst einen Abstecher in die Welt der Betriebssysteme und Rechnerarchitekturen. Unsere Betrachtungen an dieser Stelle dienen dem besseren Verständnis aller Rechnerarchitekturen – wobei es keine Rolle spielt, ob es sich dabei um einen realen oder virtuellen Server, um eine reale oder virtuelle Netzkomponente (Router, Switch), um einen realen oder virtuellen Speicher (Storage) oder um einen realen oder virtualisierten Desktop (Client) handelt. Die Ausführungen in diesem Kapitel basieren auf dem leider vergriffenen Buch "Betriebssysteme" [3].

## 1.1 Arten von Betriebssystemen

In diesem Kapitel wird der Begriff „Betriebssystem“ (engl. operating system) erläutert, wie er in der Literatur von verschiedenen Organisationen und Autoren erklärt wird. In einem zweiten Teil werden die Betriebssysteme nach diversen Kriterien klassifiziert.

### 1.1.1 Definition des Begriffs „Betriebssystem“

Grundsätzlich versteht man unter diesem Begriff Software, die zusammen mit den Hardwareeigenschaften des Computers die Basis zum Betrieb eines Computers bildet. Insbesondere steuert und überwacht ein Betriebssystem die Abarbeitung von Programmen und stellt eine Schnittstelle zur Anwendersoftware und somit zum Benutzer bereit. Vereinfacht gesagt macht erst das Betriebssystem

die Benutzung des Computers möglich. Moderne Betriebssysteme stellen dem Benutzer grafische Benutzeroberflächen zur Verfügung, so dass dieser im Gegensatz zu den Anfängen der Informatik überhaupt nichts mehr vom Aufbau der Computer Hardware verstehen muss.

### **1.1.2 Definitionen aus der Literatur**

Einige Definitionen zu den Betriebssystemen, wie sie in der Literatur gefunden werden können, sind hier aufgeführt. Eine genaue Erklärung aller Begriffe im Zusammenhang mit Computertechnik und Computernetzwerken findet sich in der International gültigen ISO-Norm 2382<sup>1</sup> mit den vorläufig veröffentlichten Teilen 1 bis 34.

Die Norm DIN 44300 ist ein deutscher Auszug aus der ISO Norm 2382. Ein Teil der deutschen Literatur über Betriebssysteme referenziert diese Norm an Stelle der ISO-Norm. In dieser Norm steht zum Begriff „Betriebssystem“:

*“Die Programme eines digitalen Rechensystems, die zusammen mit den Eigenschaften dieser Rechenanlage die Basis der möglichen Betriebsarten des digitalen Rechensystems bilden, und die insbesondere die Abwicklung von Programmen steuern und überwachen.”*

Im DUDEN-Informatik findet sich die Definitionen zum Begriff „Betriebssystem“ wie folgt:

*“Zusammenfassende Bezeichnung für alle Programme, die die Ausführung der Benutzerprogramme, die Verteilung der Betriebsmittel auf die einzelnen Benutzerprogramme und die Aufrechterhaltung der Betriebsart steuern und überwachen.”*

Das folgende Bild 1 zeigt, wie ein Benutzer das Anwenderprogramm (z.B. einen Texteditor) benutzen kann. Damit dies möglich wird, muss das Anwenderprogramm auf einem Betriebssystem laufen. Das Betriebssystem ermöglicht somit dem Anwenderprogramm die Nutzung der Hardware.

## **1.2 Computer Hardware**

In diesem Kapitel wird anhand des Rechnermodelles des Herrn von Neumann gezeigt, wie der Begriff „Hardware“ im Zusammenhang mit den Betriebssystemen definiert ist. Aus Gründen der Verständlichkeit wird vor allem auf die Hardware der Datenverarbeitungs-Maschinen eingegangen. Die beschriebenen Begriffe und Konzepte gelten jedoch auch für andere digitale Geräte, die mit einer Recheneinheit (CPU) ausgestattet sind<sup>2</sup>. Das genaue Verständnis der grundlegenden Hardwarearchitektur und deren Zusammenspiel mit dem Betriebssystem ist sinnvoll, weil dadurch klar wird, welche komplexen Aufgaben die Virtualisierung zu erfüllen hat.

### **1.2.1 Von Neumann Modell**

Eines der ersten Rechnermodelle war das Modell von John von Neumann. Es wurde 1946 von ihm, Herman H. Goldstine und Arthur W. Burks, in der Schrift „Preliminary discussion of the logical design of an electronic computing instrument“ [4] als US Army Ordnance Department Report erstmalig vorgestellt. Der Report beschreibt die heute noch bei Computersystemen vorherrschende Rechner-Architekturform, von der erst in letzter Zeit abgewichen wird, da andere Rechnerkonzepte geeignetere Eigenschaften aufweisen.<sup>3</sup>

### **Beschreibung des von Neumann Rechners**

Die drei Verfasser dieses Reports haben sich zum Ziel gesetzt, eine Maschine zu beschreiben, die im wesentlichen jede Rechenaufgabe

lösen kann.

Die Rechenaufgabe besteht aus Anweisungen (Befehlen) und Daten (z.B. die zu berechnenden Zahlenwerte). Sie soll vom Benutzer über eine Ein- / Ausgabe-Schnittstelle in ein Speicherwerk eingelesen und von dort einem Rechenwerk zugeführt werden können.

Das Rechenwerk soll die notwendigen Rechenoperationen (z.B. plus und minus), logische Verknüpfungen (z.B. AND, OR) und Transformationen (z.B. Dezimal-Komma verschieben, Runden) ausführen.

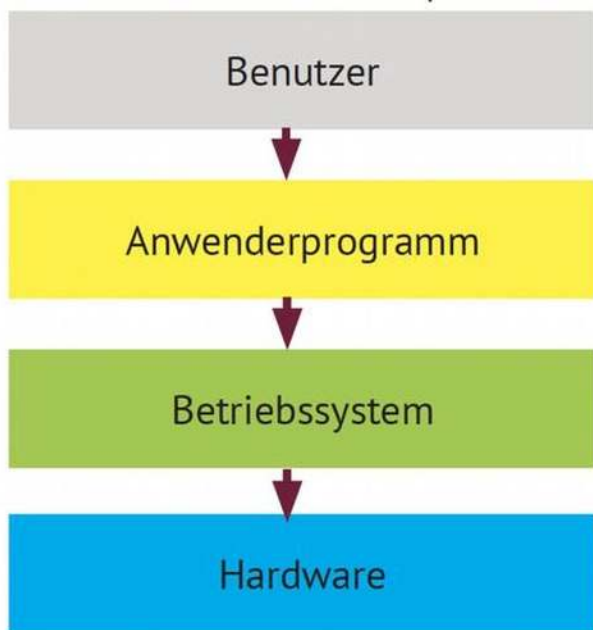


Bild: 1 Benutzer / Anwendung / Betriebssystem / Hardware

Ein Steuerwerk (Leitwerk) soll den Ablauf der Anweisungen überwachen und steuern. Sowohl die Rechenanweisungen (Programme) wie auch die Ein- und Ausgabedaten sind in einem

Speicherwerk eingelagert und können nach Bedarf ein- oder ausgelesen werden.

Die 4 wesentlichen Grundelemente der „von Neumann“-Rechenmaschine sind:

- das Rechenwerk, zuständig für arithmetische, logische und transformierende Operationen. (ALU, Arithmetic Logic Unit)
- das Steuerwerk, zuständig für die Kontrolle und den Ablauf der binär verschlüsselten Anweisungen (Befehle). Die Anweisungen (Befehle) werden sequentiell (nacheinander) abgearbeitet. Die Kontrollkomponente ist für die Abarbeitung der Anweisungen (Befehle) verantwortlich. (CU, Control Unit)
- das Speicherwerk, zuständig für die Speicherung der Daten und Programme. Anweisungen (Befehle) und Daten befinden sich gemeinsam im Speicher. (MU, Memory Unit)
- das Ein - / Ausgabewerk, zuständig für die Kommunikation mit dem Benutzer, respektive mit den Peripheriegeräten, wie z.B. Drucker oder Tastaturen und Bildschirme. Es dient der Ein- und Ausgabe von Anweisungen und Daten. (IOU, Input Output Unit)

Das Steuer- und Rechenwerk werden unter dem Begriff „Prozessor“ zusammengefasst.



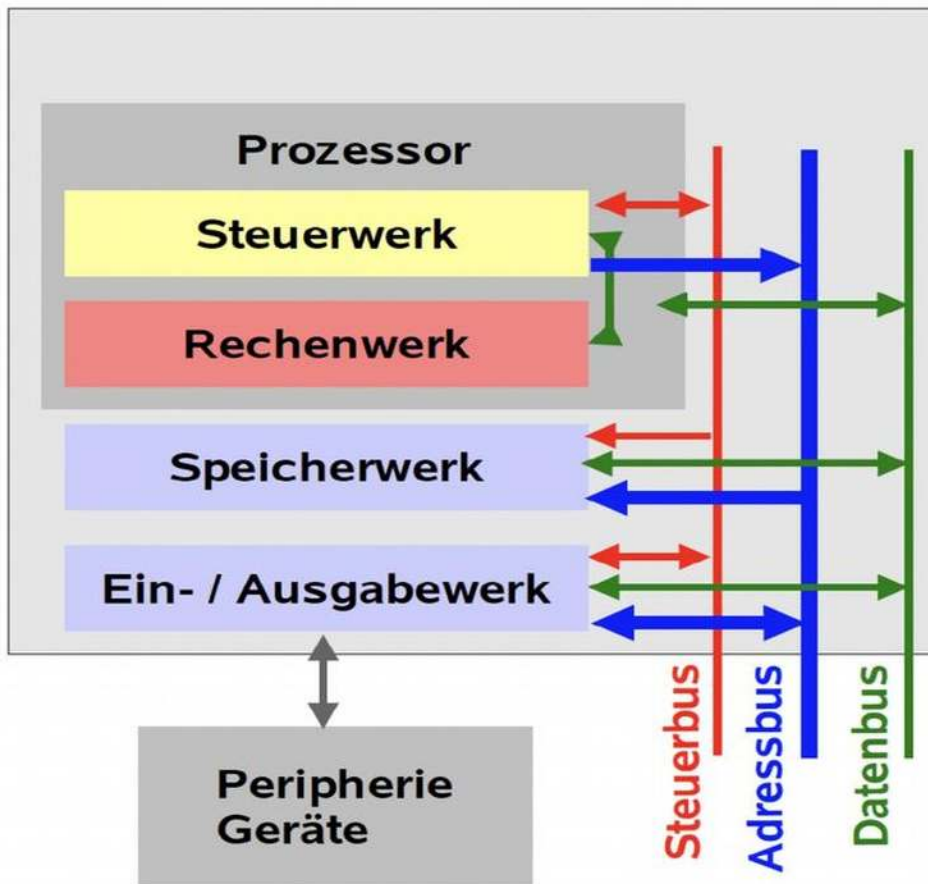


Bild: 2 Die CPU nach von Neumann

Alle 4 Werke (Steuer-, Rechen-, Speicher- und Ein- / Ausgabewerk) bilden zusammen die Zentraleinheit eines Computers. Diese Zentraleinheit wird in Englisch CPU, central processing unit genannt. Bild 2 zeigt eine Rechenmaschine nach von Neumann.

Aus praktischen Gründen wird heute das Rechen- und Steuerwerk in einem Prozessor zusammengefasst<sup>4</sup> und das Speicherwerk bestehend aus ROM (Read Only Memory) und RAM, (Random Access Memory) in separaten Chip untergebracht.

Der Informationsaustausch zwischen den verschiedenen Elementen der von Neumann Maschine geschieht über interne Datenleitungen. Diese nennt man „Bus“. Damit sich die Steuersignale, die Daten und

Befehle sowie die Speicheradressen nicht in die Quere kommen, werden keine direkten Datenleitungen verwendet, sondern getrennte Busse. Diese Busse heissen, den transportierten Datenarten entsprechend, Adressbus, Steuerbus und Datenbus. In Bild 2 sind die Busse gezeigt und es ist ersichtlich, welche Elemente in welchen Richtungen mit den Bussen kommunizieren.

### Eigenschaften des von Neumann Rechners

Möchte man nun mit einer solchen Maschine rechnen, so müssen noch einige weitere grundsätzliche Eigenheiten bekannt sein.

1. Eine CPU versteht nur Anweisungen in binärer Form. Alle Anweisungen und Daten müssen somit in „Worte“ von 8, 16, 32 oder neuerdings 64 Bit Länge umgewandelt werden.[5](#)
2. Jede CPU „versteht“ nur eine ganz bestimmte Anzahl von speziellen Maschinen-Befehlen (instructions). Da diese Befehle alle in binärer Form vorliegen müssen (eine Folge von 8 ... 64 Einsen und Nullen), hat man sich besser merkbare Namen für diese Maschinenbefehle ausgedacht. Diese Sprache nennt man Mnemonic, ein englisches Wort, das etwa „Gedächtnisstütze“ bedeutet.

Als Beispiel soll in einem Game Boy Gerät mit einem erweiterten, auf dem Z80 basierenden 8080 8-Bit-Prozessor der Zahlenwert 5 in ein Register (Speicher) geschrieben werden[6](#). Das Mnemonic für <Load Data to/from Registers/Memory> ist LD. RP bezeichnet bei diesem Prozessor einen 16 Bit Speicher (oder besser ein Speicherpaar aus zwei 8-Bit-Speichern). Mit # (data) wird der Wert, der in das Register geschrieben werden soll, bezeichnet. Der Befehl lautet somit LD RP,#5. Die Mnemonic-Programmanweisungen müssen nun von einem ASSEMBLER-Systemprogramm[7](#) in die gewünschten

Maschinenbefehle umgewandelt werden.

Der Zusammenhang zwischen Mnemonic und Maschinensprache wird am besten durch das folgende Beispiele erklärt: Das Mnemonic LD RP,#5 besteht aus einem Befehl (LD RP) und Daten (5) und wird durch das Assemblieren in 8-Bit-Folgen in der Maschinensprache umgewandelt; 0011'0010 für den Befehl (LD RP) und 0000'0000'0000'0101 für die Daten (5) im zweimal 8-Bit-Speicherpaar (RP).

3. Der Speicher ist entsprechend der Wortlänge der Maschine in 8, 16, 32 oder 64 Bit breite Speicherzellen eingeteilt. Jede Zelle hat eine Adresse.
4. Im Speicherwerk befinden sich sowohl Daten (z.B. Zahlenwerte) als auch Anweisungen. Aus diesem Grund muss das Steuerwerk gut „aufpassen“, dass jeweils die richtige Art der Speicherinhalte dem Rechenwerk übergeben wird.
5. Die Abarbeitung der Speicherinhalte geschieht in zwei Schritten:
  1. Schritt: Der Speicherplatzinhalt aus Zelle 1 wird gelesen, als Anweisung interpretiert und dem Rechenwerk zur Ausführung übergeben<sup>8</sup>.
  2. Schritt: Hier wird der Speicherplatzinhalt der durch den Befehl spezifizierten Adresse (in unserem Beispiel oben mit dem Befehl ADD z.B. Platz 10 oder 11) gelesen und dem Rechenwerk zur Bearbeitung (im Beispiel Addieren) übergeben<sup>9</sup>.
6. Es ist keine Parallel-Bearbeitung mehrerer Befehle oder Daten möglich. Ein von Neumann-Rechner bearbeitet zu jedem Zeitpunkt immer nur einen Befehl (sequenziell), der immer eine Datenoperation im Rechenwerk bewirkt. Dies ist auch als von Neumannscher Flaschenhals bekannt. Eine Abhilfe dafür bieten Parallel-Rechner, indem solche Rechner-Architekturen ein

paralleles Abarbeiten von Befehlen aus dem Speicher ermöglichen.

7. Es existiert nur ein Verbindungsweg zwischen dem Prozessor und dem Speicherwerk. Es lässt sich somit immer nur ein Wort auf einmal über die Verbindungen transportieren.

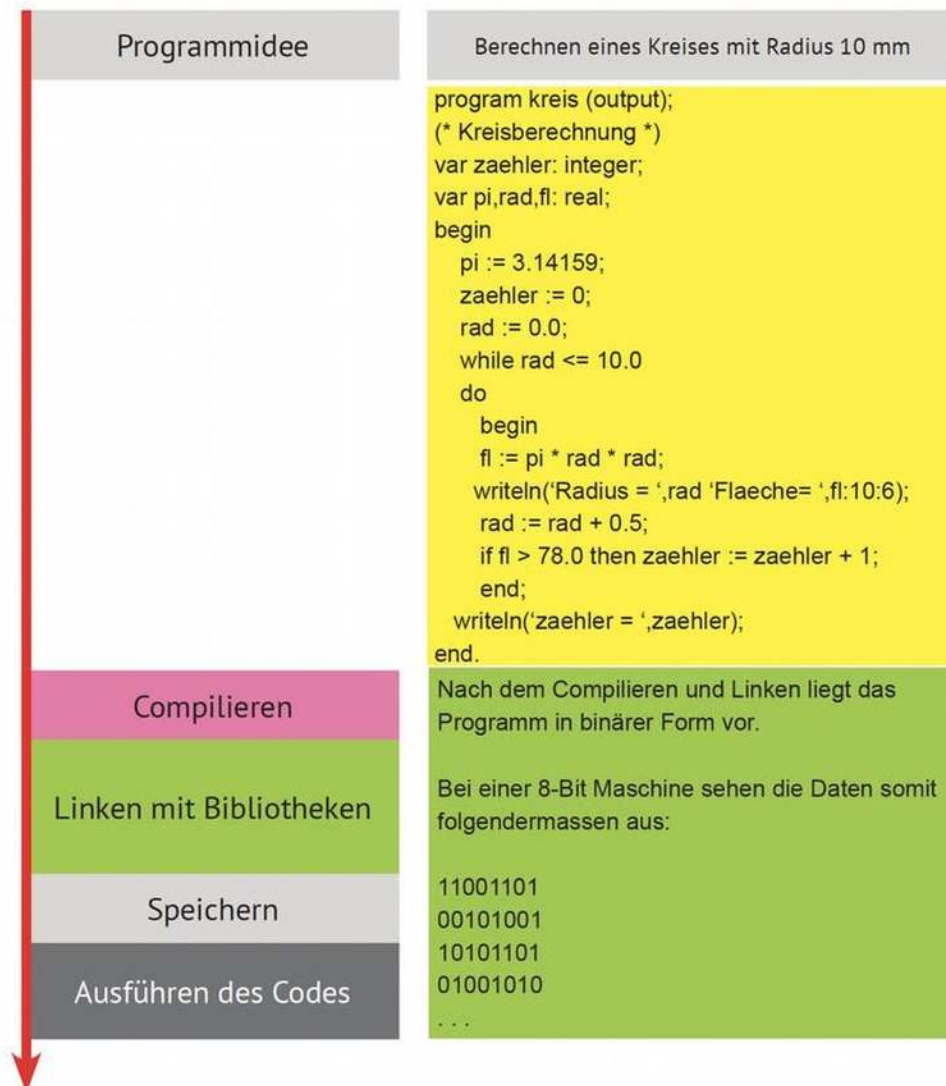


Bild: 3 Von der Programmidee zur Applikation

## Benutzung des von Neumann Rechners

Möchte nun ein Benutzer eines von Neumann Rechners (oder auch eines modernen PCs) eine Aufgabe<sup>[10](#)</sup> lösen, muss er zuerst in einer Programmiersprache mit Hilfe eines Editors<sup>[11](#)</sup> eine Abfolge von

Befehlen programmieren<sup>12</sup>. Er muss auch festlegen, woher beispielsweise die Eingabewerte für die Berechnung von Ausgabewerten stammen sollen<sup>13</sup>.

Anwendersoftware wird von speziell ausgebildeten Programmierern erstellt. Es besteht eine klare Trennung zwischen den Programmierern und den Benutzern eines Rechners, respektive der Anwendersoftware. Dies nicht nur, weil die Benutzer in der Regel weder die Hardware noch das Programmieren verstehen, sondern auch aus Gründen der Sicherheit. Es wäre sonst zum Beispiel möglich, dass ein Benutzer einer Bank-Applikation im Programm, das er anschliessend selber nutzen soll, einen „Fehler“ einprogrammiert, der alle Rundungsdifferenzen aus Banktransaktionen seinem Konto gutschreibt!

Programme werden heute nicht unbedingt in der maschinennahen Programmiersprache Mnemonic geschrieben, sondern in höher entwickelten und somit einfacher zu bedienenden Sprachen wie C++, Pascal, Delphi und anderen. Diese Programme müssen zuerst in Maschinensprache umgewandelt werden. Dies geschieht mit einem sogenannten Compiler<sup>14</sup>. Diese wandeln die Programmanweisungen in Maschinensprache um. Für jeden Prozessor-Typ wird dabei ein charakteristischer Befehlssatz in binärer Form erzeugt<sup>15</sup>.

Damit man nicht immer alle Standardbefehlsfolgen, wie sie zum Beispiel für die Bildschirmausgabe (in Bild 3: `writeln('zaehler =', zaehler);`), das Einlesen der Mausklicks und Ähnliches vorkommen, selber programmieren muss, werden diese Programmteile einmal geschrieben und in Bibliotheken (libraries) zusammengefasst.