

Grundlagen der Systemsoftware (GSS)

Prof. Dr. Bernd E. Wolfinger ¹⁾ [Teil I: Systemsoftware]

sowie

Prof. Dr. Hannes Federrath ²⁾ [Teil II: Sicherheit]

1) Telekommunikation & Rechnernetze (TKRN)

2) Sicherheit in Verteilten Systemen (SVS)

Fachbereich Informatik, MIN-Fakultät, Universität Hamburg

INHALT:

B.Sc.-Modul “**Grundlagen der Systemsoftware**”

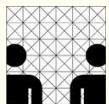
- Übersicht über Grundkonzepte, Grundbausteine und Architekturen von Systemsoftware
- Grundlagen und Grundkonzepte von Betriebssystemen
- Grundkonzepte der Nebenläufigkeit und Verteilung
- Grundkonzepte der Vernetzung

Teil I : B.E. Wolfinger

- Einführung in die Agententechnologie

- Einführung in die IT-Sicherheit
- Authentication & Authorization
- Sicherheitsmodule
- Sicherheitsprotokolle
- Sicherheit von Betriebssystemen

Teil II : H. Federrath

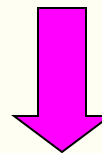


INHALT:

B.Sc.-Modul “Grundlagen der Systemsoftware”

Teil I (Systemsoftware) : B.E. Wolfinger

- Übersicht über Grundkonzepte, Grundbausteine und Architekturen von Systemsoftware
- Grundlagen und Grundkonzepte von Betriebssystemen
- Grundkonzepte der Nebenläufigkeit und Verteilung
- Grundkonzepte der Vernetzung



abgebildet auf
2 Bereiche

A. Betriebssysteme

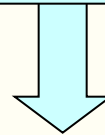
B. Rechnernetze

INHALT:

B.Sc.-Modul “Grundlagen der Systemsoftware”

Sicherheit: H. Federrath

- Einführung in die IT-Sicherheit
- Rechner- und Betriebssystem-Sicherheit
- Einführung in die Kryptographie
- Public Key Infrastructures (PKI)

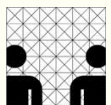


Lernziele

- Schutzziele in IT-Systemen verstehen und definieren können
- Anforderungen an Identifikations-, Zugangs- und Zugriffskontrollsysteme aufstellen und auswählen
- Gemeinsamkeiten und Unterschiede verschiedener Verschlüsselungs- und Authentikationssysteme verstehen und diese anwenden können

Literatur : *Bereiche* Betriebssysteme und Rechnernetze

- [ALS04] S. Abeck, P.C. Lockemann, J. Schiller, J. Seitz: Verteilte Informationssysteme, dpunkt-Verlag, 2004, 471 S.
- [BeG99] F. Bergmann, H. J. Gerhardt (Hrsg.): Taschenbuch der Telekommunikation, C. Hanser-Verlag 1999
- [Bra03] R. Brause: Betriebssysteme, Springer-Verlag 2003
- [Com07] D. Comer, The Internet Book: Everything you need to know about Computer Networking and how the Internet works, 4th ed., Pearson Prentice-Hall 2007, 380 S.
- [Com09] D. Comer, Computer Networks and Internets, 5th ed., Prentice-Hall 2009
- [CDK12] G. Coulouris, J. Dolimore, T. Kindberg, G. Blair: Distributed Systems: Concepts and Design, Addison Wesley, 5th Ed., 2012
- [EKR05] E. Ehses, L. Köhler, P. Riemer et al.: Betriebssysteme – Ein Lehrbuch mit Übungen zur Systemprogrammierung in UNIX/Linux, 2. Aufl., Pearson Studium 2005, 324 S.
- [Far04] A. Farrel: The Internet and its Protocols – A Comparative Approach, Morgan Kaufmann 2004
- [FaS12] K.R. Fall, W.R. Stevens: TCP/IP Illustrated, Vol. 1, 2nd ed., The Protocols, Addison-Wesley 2012, 1017 S.
- [FoM12] B.A. Forouzan, F. Mosharraf: Computer Networks, Mc. Graw-Hill 2012, 931 S.
- [Gla06] E. Glatz: Betriebssysteme, dpunkt-Verlag 2006, 736 S.
- [Gor09] W. Goralski: The Illustrated Network – How TCP/IP Works in a Modern Network, Morgan Kaufmann 2009, 797 S.
- [HaB03] T. Harris, J. Bacon: Operating Systems: Concurrent and Distributed Software Design, Pearson 2003
- [Hal05] F. Halsall: Computer Networking and the Internet, 5th ed., Addison-Wesley 2005
- [HAN99] H.-G. Hegering, S. Abeck, B. Neumair: Integriertes Management vernetzter Systeme: Konzepte, Architekturen und deren betrieblicher Einsatz, dpunkt-Verlag 1999, 607 S.
- [HPS00] H. Häckelmann, H. J. Petzold, S. Strahinger: Kommunikationssysteme – Technik und Anwendungen, Springer-Verlag 2000
- [JuW98] V. Jung, H.-J. Warnecke: Handbuch für die Telekommunikation, Springer 1998
- [Klu00] N. Klussmann: Lexikon der Kommunikations- und Informationstechnik, 2. Aufl., Hüthig-Verlag 2000
- [KrR02] G. Krüger, D. Reschke (Hrsg.): Lehr- und Übungsbuch Telematik, 2. Aufl., Fachbuchverlag Leipzig C. Hanser 2002



Literatur : Bereiche Betriebssysteme und Rechnernetze (Forts.)

- [KuR13] J. F. Kurose, K. W. Ross: Computer Networking – A Top-Down Approach, 6th ed., Pearson/Addison-Wesley 2013, 888 S. bzw. Computernetzwerke - Der Top-Down-Ansatz, 4. Aufl., Pearson /Addison-Wesley 2008, 896 S.
- [Man13] P. Mandl: Grundkurs Betriebssysteme, 3. Aufl., Teubner-Verlag 2013, 396 S.
- [MEK03] G. Müller, T. Eymann, M. Kreutzer: Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft, Oldenbourg-Verlag 2003
- [NeS01] J. Nehmer, P. Sturm: Systemsoftware – Grundlagen moderner Betriebssysteme, 2. Aufl., dpunkt-Verlag 2001
- [Nut04] G. Nutt: Operating Systems: A Modern Perspective, Addison-Wesley 2004
- [ObV06] W. Oberschelp, G. Vossen: Rechneraufbau und Rechnerstrukturen (10. Auflage), Oldenbourg-Verlag 2006, 564 S.
- [PeD08] L. L. Peterson, B. S. Davie: Computernetze, 4. Aufl., dpunkt-Verlag, 2008, 858 S.
- [Pro02] W. Proebster: Rechnernetze – Technik, Protokolle, Systeme, Anwendungen, 2. Aufl., Oldenbourg-Verlag 2002
- [SGG02] A. Silberschatz, P. Galvin, G. Gagne: Operating System Concepts, J. Wiley 2002
- [SiB06] H.-J. Siegert, U. Baumgarten: Betriebssysteme – Eine Einführung, 6. Aufl., Oldenbourg-Verlag 2006, 405 S.
- [Sie02] G. Siegmund: Technik der Netze, 5. Aufl., Hüthig-Verlag 2002
- [SMW06] R. Steinmetz, M. Mühlhäuser, M. Welzl: Rechnernetze, C. Hanser-Verlag 2006
- [Sta09] W. Stallings: Betriebssysteme – Prinzipien und Umsetzung, 4. Aufl., Prentice-Hall 2002/5, 896 S.; *engl. Original: Operating Systems: Internals and Design Principles, 6th ed., Prentice-Hall 2009*
- [Sta11] W. Stallings: Data & Computer Communications, 9th ed., Prentice-Hall 2011, 881 S.
- [Ste08] E. Stein: Rechnernetze und Internet (Taschenbuch), 3. Aufl., C. Hanser-Verlag 2008
- [Tan08] A. S. Tanenbaum: Modern Operating Systems, 3rd ed., Pearson 2008, 1076 S.; *deutsche Übersetzung (2nd ed.): Moderne Betriebssysteme, 3. Aufl., Pearson 2009, 1239 S.*
- [Tan10] A. S. Tanenbaum, D. J. Wetherall: Computer Networks, 5th ed., Prentice-Hall 2010, 951 S. *deutsche Übersetzung (4th ed.): A. Tanenbaum: Computernetzwerke, 4th ed., Prentice-Hall 2003*
- [TaW06] A. S. Tanenbaum, A.S. Woodhull: Operating Systems: Design and Implementation, 3rd ed., Pearson 2006, 1080 S.
- [Zen01] A. Zenk: Lokale Netze – Die Technik fürs 21. Jahrhundert, 2. Aufl., Addison-Wesley 2001

INHALT: *Bereiche A* "Betriebssysteme" & *B* "Rechnernetze"

-
- A1. Betriebssysteme: Einführung und Motivation
 - A2. Prozesse: Scheduling und Betriebsmittelzuteilung
 - A3. Prozesse: Synchronisation und Kommunikation
 - A4. Speicherverwaltung
 - A5. Dateisysteme
 - A6. Ein-/Ausgabe
-
- B1. Kommunikations- und Rechnernetze: Einführung und Motivation
 - B2. Einige Gemeinsamkeiten von Betriebssystemen und Rechnernetzen, Architekturmodelle und Diensthierarchien
 - B3. Übertragungsmedien und Signalübertragung
 - B4. Datenübertragung über Leitungen und drahtlose Verbindungen
 - B5. Lokale Rechnernetze
 - B6. Vermittlungsnetze und IP
 - B7. Transportdienste und -protokolle
 - B8. Realisierungsaspekte für Kommunikationssoftware und kurzer Ausblick

Vorlesung [3 SWS] → B.E. Wolfinger (im April/Mai) & H. Federrath

Mo, 10:15 – 11:45 Uhr; Phil-B sowie

Fr, 10:15 – 11:45 Uhr; ErzWiss H (Fr. 14-tägig im April und Mai, d.h. im „Wolfinger-Teil“ – Ausnahme: GSS-Vorl. findet statt am Fr, den 12.4.'13; evtl. Fr. nicht strikt 14-tägig im Juni/Juli) → Liste der Freitagstermine im April/Mai '13: 5.4.,12.4.,19.4.,3.5.,17.5.,31.5.

Übungen [14-tägig, je 2 „h“ (d.h. 90 min)]

➤ **Beginn** in 16. Kalender-Woche, d.h. ab **Di., 2. April 2013**

- Aufteilung in 14 Übungsgruppen (Termine, vgl. STiNE)
- 1. Übungsblatt bereits in 1. Vorl. verteilt sowie in STiNE verfügbar

Übungstermine im April/Mai: Ü0: 1.-5.4.(„Kennenlern-Ü-Gruppen“), Ü1: 15.-19.4., Ü2: 29.4.-3.5., Ü (1.5.) verlegt auf 8.5.! Ü3: 13.5.-17.5., Ü4: 27.5.-31.5.

➤ *Nota bene: Übungen sind wichtig (... und die erfolgreiche Teilnahme ist Voraussetzung für die Teilnahme an der Klausur) !!!*

➤ hinreichende **Anforderung für Übungsschein:**

- Teilnahme an ≥ 5 von 7 Übungen
- ≥ 50 % korrekt bearbeitete Aufgaben bezogen auf jeden der beiden Teile der Vorlesung [Übungsaufgaben können generell in Teams von 1 bis max. 4 Studierenden bearbeitet werden !].

Prüfungsmodalitäten für GSS (im Bachelor-Studiengang!)

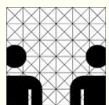
Bachelor-Modul 'Grundlagen der Systemsoftware', SoSe 2013

- Umfang 3 SWS Vorlesung + 1 SWS Übungen
- Erfolgreiche Teilnahme an Übungen ist *Voraussetzung für Zulassung zu Klausur*
- Art der Prüfung: schriftliche Klausur

Klausur direkt nach Ende des SoSe (d.h. ab ca. Mitte Juli '13)

(zu exaktem Klausurtermin, siehe Web-Seiten des Fachbereichs Informatik der Uni HH,

zu Prüfungsanmeldungen siehe Infos des FBI-Studienbüros).



Allgemeine Quellenangaben zu den benutzten Folien

Basis für die GSS-Vorlesung im „Wolfinger-Teil“:

Aktualisierte Folien aus den ersten beiden GSS-Zyklen im SoSe 2007 und 2008 (GSS-Vorlesung, abgesehen von Sicherheitsteil, wurde von „BEW“ konzipiert)

Weitere Quellen:

- Frühere Vorlesungen von Dr. M. Lehmann zu Betriebssystemen
- Einige Abbildungen aus Lehrbüchern → daher Folien nur für persönlichen Gebrauch bestimmt !
- Einige Folien von Prof. W. Lamersdorf aus der GSS-Vorlesung des SoSe 2012

Zur ersten Motivation für und Einstimmung auf die GSS-Themen

Jüngere Beispiele:

Hannover - Die weltgrößte Computermesse CeBIT in Hannover. Im Mittelpunkt standen dort unter anderem wiederum **schnelle Internet-Dienste** für unterwegs, das **Microsoft-Betriebssystem Windows 7**, **Fernsehen über das Internet** und auch **Unterhaltungselektronik** wie flache TV-Geräte Mobilgeräte etc..

2009/10 speziell: u.a. **Mobile Systeme (Tablets!)**, „**Internet der Dinge**“ und (BS:) **Windows 7**, **Android** etc.

2012: **Cloud Computing (!)**, **Security (!)**

Top-Thema der Cebit 2011 „Work and Life with the Cloud“: **Cloud-Services** sind aktuell einer der wichtigsten Wachstumstreiber der internationalen IT-Industrie. Die CeBIT 2011 widmet sich mit „Work and Life with the Cloud“ diesem Zukunftsmarkt und demonstriert an konkreten Beispielen wie Cloud-Konzepte die Arbeits- und Lebenswelten der Menschen verändern. [<http://stadtleben.de/gmbh/>], Auf der Cebit 2012 kamen speziell **Sicherheitsaspekte** als Schwerpunkt dazu.

Zur Zukunft des Internet ?!

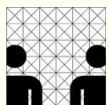
[**Thesen aus: 10 Fool-proof Predictions for the Internet in 2020**
Network World (01/04/09) Marsan, Carolyn Duffy]

PART I of list

Network World offers **10 “surefire bets” about what the Internet will look like in 10 years**. They include:

- 1.** As computer scientists work to improve the Internet's design, the global network is expected to change dramatically over the next 10 years. The Internet currently has about 1.7 billion users, but the U.S. National Science Foundation (NSF) expects the Internet will have **nearly 5 billion users by 2020**.
- 2.** The Internet also will be **more geographically dispersed** in 10 years, spreading to more developing regions.
- 3.** Ten years from now, the Internet will be a **network of things**, not computers. Today, the Internet has approximately 575 million host computers, but the NSF expects infrastructure sensors alone to surpass the number of host computers by several orders of magnitude.
- 4.** The Internet also will **carry more content**. Cisco estimates that global Internet traffic will increase to **about 44 exabytes** ($= 44 \times 10^{18}$ Bytes) **per month by 2012**.
- 5.** In 2020, the **Internet will be wireless**. In the second quarter of 2009, the number of mobile subscribers hit 257 million, representing an 85 percent increase year-over year for high-speed data networking technologies. **By 2014, approximately 2.5 billion people** will subscribe to **some form of mobile broadband**, according to Informa.

... **6. - 10.** cf. **PART II** of list



Zur Zukunft des Internet ?!

[**Thesen aus:** 10 Fool-proof Predictions for the Internet in 2020
Network World (01/04/09) Marsan, Carolyn Duffy]

PART II of list

Network World offers 10 “surefire bets” about what the Internet will look like in 10 years.

They include:

1. - 5. cf. **PART I** of list

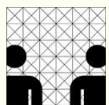
... **6.** More services will use **cloud computing**. The NSF is encouraging researchers to develop better ways to map users and information in a cloud-computing infrastructure.

7. Ten years from now, the Internet also will be **greener**. Future Internet architecture needs to be **more energy efficient**, as the amount of energy used by the Internet doubled between 2000 and 2006, according to the Lawrence Berkeley National Laboratory.

8. Network management will be **more automated** in 2020. The NSF is researching new network management tools for the future Internet, including **automated reboot systems, self-diagnosis protocols, finer-grained data collection, and better event tracking**.

9. The Internet will not rely on **constant connectivity**. Researchers are studying communication techniques that can handle delays or easily forward information to different users.

10. The Internet will **attract more hackers**, and computer scientists will work to make it more secure.



... weitere Motivation für die GSS-Themen (u.a. Internet)

Video Road Hogs Stir Fear of Internet Traffic Jam

New York Times (03/13/08) P. A1; Lohr, Steve

The increasing visual richness of online communications and entertainment, including video clips and movies, social networks, and multiplayer games, poses a threat that could cause massive Internet traffic delays, researchers warn. *Some analysts estimate that YouTube consumed as much bandwidth in 2007 than the entire Internet did in 2000.* In a report published last November, Nemertes Research projected that Internet demand could outpace network capacity by 2011. However, even those most concerned over the Internet traffic surge say it poses more of a challenge than an impending catastrophe, and most are not predicting a lights-out Internet crash. Instead, they warn that Internet users could experience sluggish download speeds and difficulty with data-heavy services. Nemertes President Johna Till Johnson says the Internet will not collapse, but there will be a growing category of tasks that will no longer be able to be completed over the internet. Johnson anticipates that *Internet demand will grow by 100 percent of more per year.* Others expect that *Internet growth will more likely be about 50 percent per year.* Meanwhile, analysts note that routers are getting faster, fiber-optic transmissions are improving, and the software used for managing data packets is getting smarter. Nevertheless, experts agree that if American investment doesn't keep up, the country risks falling behind countries that have made higher-speed Internet access a priority.

... weitere Motivation für die GSS-Themen (u.a. Mobilfunk und Internet)

Schnelles Internet auf dem Handy steht vor dem Durchbruch *(Meldung vom März 2008, aktualisiert)*

Hannover (dpa) - Die Mobilfunk-Branche rüstet sich für einen baldigen Durchbruch des **mobilen Internets mit schnellen Datenverbindungen**. Zum Jahr 2012 rechnet man mit **1,8 Milliarden Nutzern mobiler Breitband-Dienste**, sagte der Chef der Telekom-Mobilfunktochter T-Mobile, Hamid Akhavan, am Donnerstag auf der Computermesse CeBIT in Hannover. Die Betreiber nehmen im Jahr **2010 erste Netze der nächsten Generation** in Betrieb, die Übertragungsgeschwindigkeiten von bis zu 100 Megabit pro Sekunde erlauben – 100 Mal schneller als die einfachste DSL-Leitung. Schon in wenigen Jahren werden *mehr Menschen mobile Breitband-Internetleitungen nutzen als DSL*.

INHALT:

Bereich A "Betriebssysteme"

- A1. Betriebssysteme: Einführung und Motivation
- A2. Prozesse: Scheduling und Betriebsmittelzuteilung
- A3. Prozesse: Synchronisation und Kommunikation
- A4. Speicherverwaltung
- A5. Dateisysteme
- A6. Ein-/Ausgabe

TOP 10 „Betriebssysteme“

- [ALS04] S. Abeck, P.C. Lockemann, J. Schiller, J. Seitz: Verteilte Informationssysteme, dpunkt-Verlag, 2004, 471 S.
- [BeG99] F. Bergmann, H. J. Gerhardt (Hrsg.): Taschenbuch der Telekommunikation, C. Hanser-Verlag 1999
- [Bra03] R. Brause: Betriebssysteme, Springer-Verlag 2003
- [Com07] D. Comer, The Internet Book: Everything you need to know about Computer Networking and how the Internet works, 4th ed., Pearson Prentice-Hall 2007, 380 S.
- [Com09] D. Comer, Computer Networks and Internets, 5th ed., Prentice-Hall 2009
- [CDK12] G. Coulouris, J. Dolimore, T. Kindberg, G. Blair: Distributed Systems: Concepts and Design, Addison Wesley, 5th Ed., 2012
- [EKR05] E. Ehses, L. Köhler, P. Riemer et al.: Betriebssysteme – Ein Lehrbuch mit Übungen zur Systemprogrammierung in UNIX/Linux, 2. Aufl., Pearson Studium 2005, 324 S.
- [Far04] A. Farrel: The Internet and its Protocols – A Comparative Approach, Morgan Kaufmann 2004
- [FaS12] K.R. Fall, W.R. Stevens: TCP/IP Illustrated, Vol. 1, 2nd ed., The Protocols, Addison-Wesley 2012, 1017 S.
- [FoM12] B.A. Forouzan, F. Mosharraf: Computer Networks, Mc. Graw-Hill 2012, 931 S.
- [Gla06] E. Glatz: Betriebssysteme, dpunkt-Verlag 2006, 736 S.
- [Gor09] W. Goralski: The Illustrated Network – How TCP/IP Works in a Modern Network, Morgan Kaufmann 2009, 797 S.
- [HaB03] T. Harris, J. Bacon: Operating Systems: Concurrent and Distributed Software Design, Pearson 2003
- [Hal05] F. Halsall: Computer Networking and the Internet, 5th ed., Addison-Wesley 2005
- [HAN99] H.-G. Hegering, S. Abeck, B. Neumair: Integriertes Management vernetzter Systeme: Konzepte, Architekturen und deren betrieblicher Einsatz, dpunkt-Verlag 1999, 607 S.
- [HPS00] H. Häckelmann, H. J. Petzold, S. Strahinger: Kommunikationssysteme – Technik und Anwendungen, Springer-Verlag 2000
- [JuW98] V. Jung, H.-J. Warnecke: Handbuch für die Telekommunikation, Springer 1998
- [Klu00] N. Klusmann: Lexikon der Kommunikations- und Informationstechnik, 2. Aufl., Hüthig-Verlag 2000
- [KrR02] G. Krüger, D. Reschke (Hrsg.): Lehr- und Übungsbuch Telematik, 2. Aufl., Fachbuchverlag Leipzig C. Hanser 2002

Literatur : Bereiche Betriebssysteme und Rechnernetze (Forts.)

- [KuR13] J. F. Kurose, K. W. Ross: Computer Networking – A Top-Down Approach, 6th ed., Pearson/Addison-Wesley 2013, 888 S. bzw. Computernetzwerke - Der Top-Down-Ansatz, 4. Aufl., Pearson /Addison-Wesley 2008, 896 S.
- [Man13] P. Mandl: Grundkurs Betriebssysteme, 3. Aufl., Teubner-Verlag 2013, 396 S.
- [MEK03] G. Müller, T. Eymann, M. Kreutzer: Telematik- und Kommunikationssysteme in der vernetzten Wirtschaft, Oldenbourg-Verlag 2003
- [NeS01] J. Nehmer, P. Sturm: Systemsoftware – Grundlagen moderner Betriebssysteme, 2. Aufl., dpunkt-Verlag 2001
- [Nut04] G. Nutt: Operating Systems: A Modern Perspective, Addison-Wesley 2004
- [ObV06] W. Oberschelp, G. Vossen: Rechneraufbau und Rechnerstrukturen (10. Auflage), Oldenbourg-Verlag 2006, 564 S.
- [PeD08] L. L. Peterson, B. S. Davie: Computernetze, 4. Aufl., dpunkt-Verlag, 2008, 858 S.
- [Pro02] W. Proebster: Rechnernetze – Technik, Protokolle, Systeme, Anwendungen, 2. Aufl., Oldenbourg-Verlag 2002
- [SGG02] A. Silberschatz, P. Galvin, G. Gagne: Operating System Concepts, J. Wiley 2002
- [SiB06] H.-J. Siegert, U. Baumgarten: Betriebssysteme – Eine Einführung, 6. Aufl., Oldenbourg-Verlag 2006, 405 S.
- [Sie02] G. Siegmund: Technik der Netze, 5. Aufl., Hüthig-Verlag 2002
- [SMW06] R. Steinmetz, M. Mühlhäuser, M. Welzl: Rechnernetze, C. Hanser-Verlag 2006
- [Sta09] W. Stallings: Betriebssysteme – Prinzipien und Umsetzung, 4. Aufl., Prentice-Hall 2002/5, 896 S.; *engl. Original: Operating Systems: Internals and Design Principles, 6th ed., Prentice-Hall 2009*
- [Sta11] W. Stallings: Data & Computer Communications, 9th ed., Prentice-Hall 2011, 881 S.
- [Ste08] E. Stein: Rechnernetze und Internet (Taschenbuch), 3. Aufl., C. Hanser-Verlag 2008
- [Tan08] A. S. Tanenbaum: Modern Operating Systems, 3rd ed., Pearson 2008, 1076 S.; *deutsche Übersetzung (2nd ed.): Moderne Betriebssysteme, 3. Aufl., Pearson 2009, 1239 S.*
- [Tan10] A. S. Tanenbaum, D. J. Wetherall: Computer Networks, 5th ed., Prentice-Hall 2010, 951 S. *deutsche Übersetzung (4th ed.): A. Tanenbaum: Computernetzwerke, 4th ed., Prentice-Hall 2003*
- [TaW06] A. S. Tanenbaum, A.S. Woodhull: Operating Systems: Design and Implementation, 3rd ed., Pearson 2006, 1080 S.
- [Zen01] A. Zenk: Lokale Netze – Die Technik fürs 21. Jahrhundert, 2. Aufl., Addison-Wesley 2001

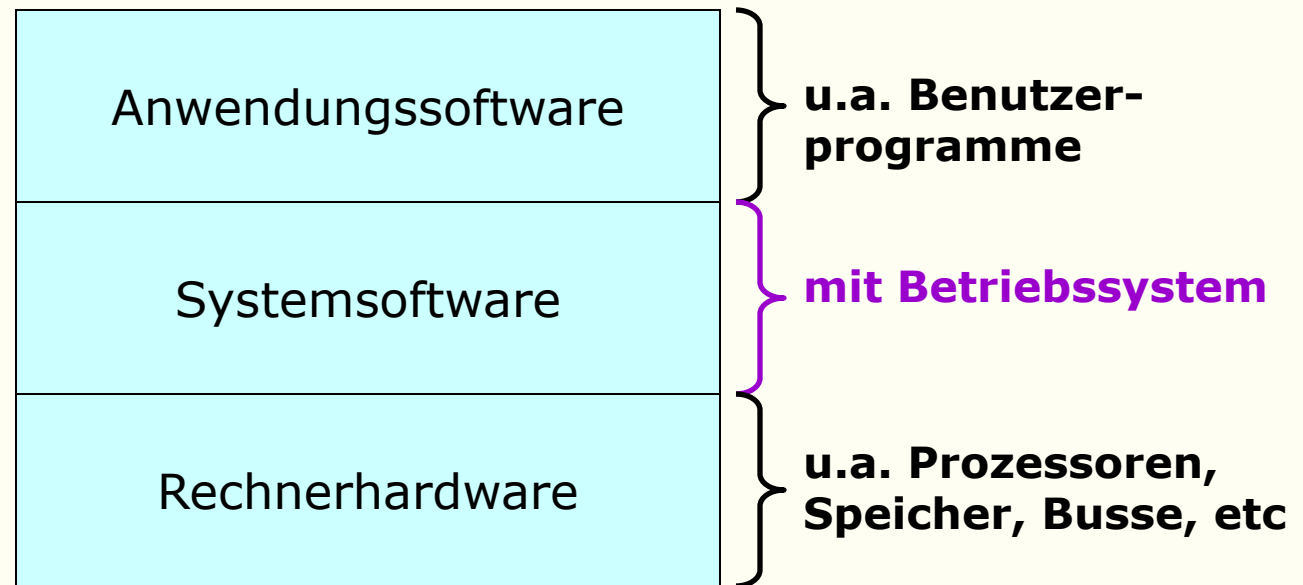
INHALT:

Bereich A "Betriebssysteme"

- A1. Betriebssysteme: Einführung und Motivation**
- A2. Prozesse: Scheduling und Betriebsmittelzuteilung
- A3. Prozesse: Synchronisation und Kommunikation
- A4. Speicherverwaltung
- A5. Dateisysteme
- A6. Ein-/Ausgabe

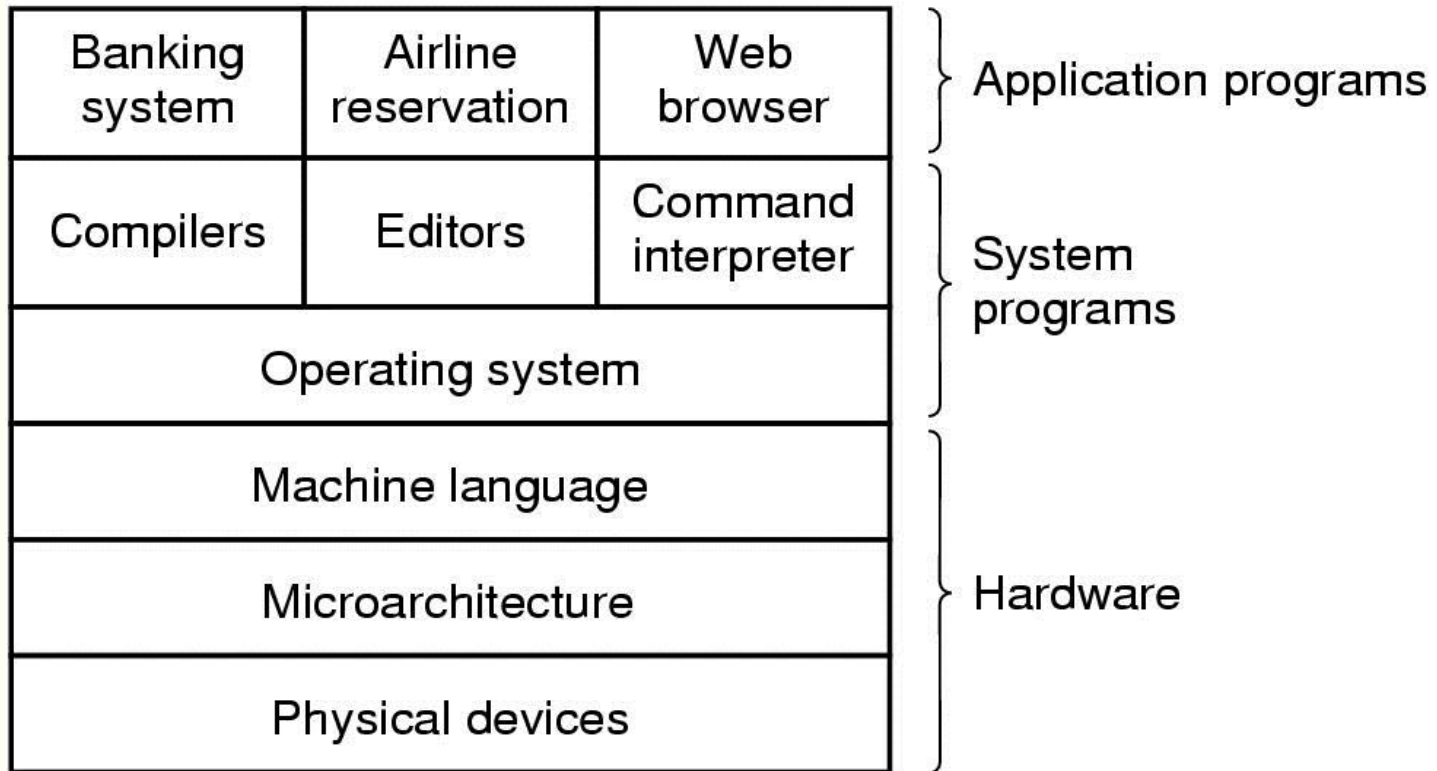
A "Betriebssysteme"

A1. Betriebssysteme: Einführung und Motivation



Elementare Hardware-/Software-Schichtenstruktur

Betriebssystem zwischen Anwendungssoftware und Hardware (Verfeinerung s. [Tan08])




- A computer system consists of
 - Application programs
 - System programs
 - Hardware

Zur Erinnerung (aus Rechnerstrukturen):

A1.1 Aufbau und Architektur konventioneller Rechensysteme

Der von Neumann-Rechner

Def. **Prozessor** (DIN 44300):

Eine Funktionseinheit innerhalb eines digitalen Rechensystems, die *Rechenwerk* und *Leitwerk* umfasst. 

wobei:

- **Rechenwerk** (arithmetic logic unit; ALU) :
Ausführung von Verarbeitungsbefehlen
auch : *Operationswerk, Verarbeitungswerk, Datenprozessor*
[Giloï 93] - W.K. Giloï: Rechnerarchitektur, Springer-Verlag, 1993
- **Leitwerk**: (control unit):
 - Steuerung der Befehlsabfolge
 - Funktions-(Op-Code-) Entschlüsselung
 - Operationssteuerung (Signale an Rechen- bzw. E/A-Werk)
auch : *Steuerwerk, Befehlsprozessor* [Giloï 93]
- **E/A-Werk**: (Input/output unit, I/O-unit):
 - Ausführung von Ein-/Ausgabebefehlen

Basiskomponenten des von Neumann-Rechners

CPU (**c**entral **p**rocessing **u**nit) \equiv Datenprozessor \cup Befehlsprozessor
auch: *Zentralprozessor, zentrale Recheneinheit* [Giloï 93]

→ bitte *nicht* : Zentraleinheit

Def. **Speicher**:

Eine Funktionseinheit innerhalb eines digitalen Rechensystems, die digitale Daten aufnimmt, aufbewahrt und abgibt. ■

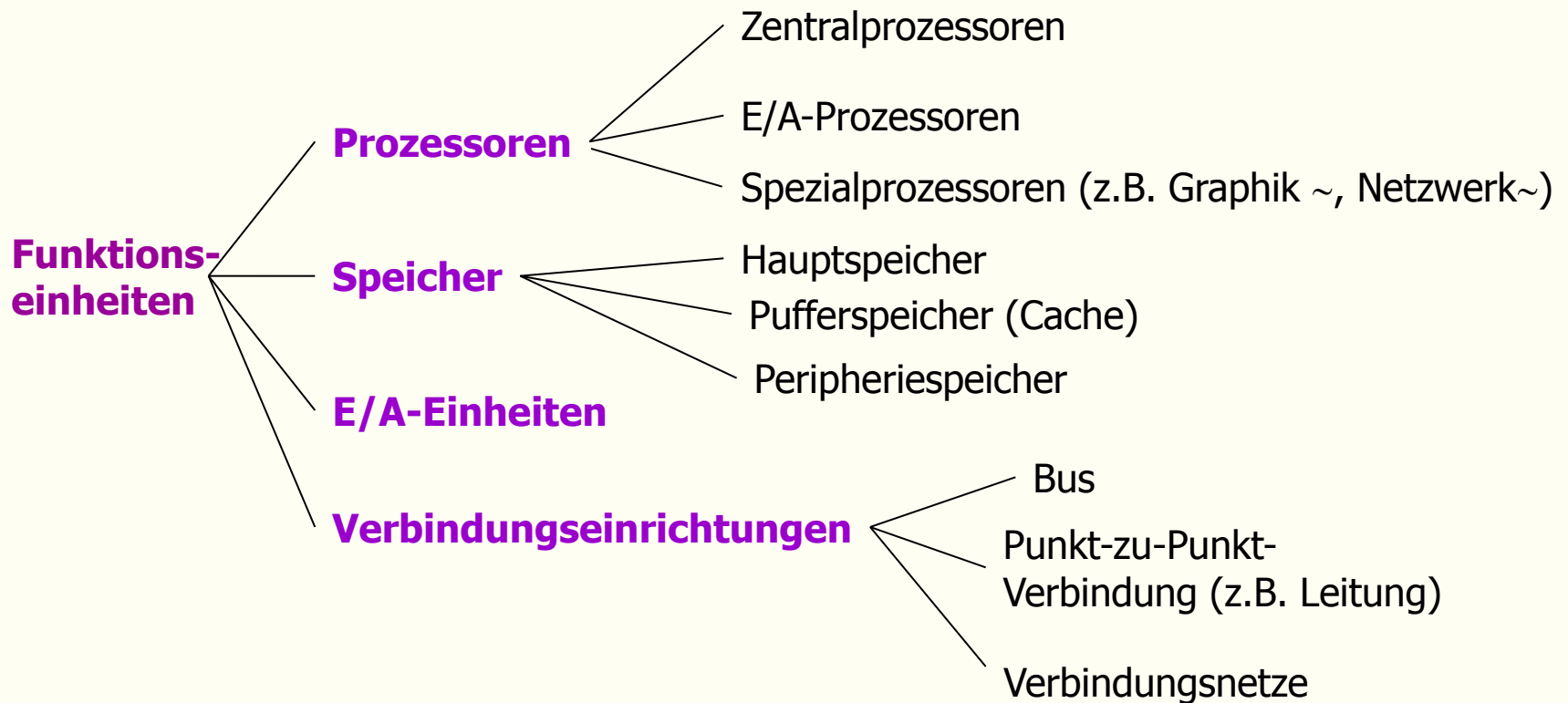
Def. **Eingabeeinheit**:

Eine Funktionseinheit innerhalb eines digitalen Rechensystems, mit der das System Daten von außen her aufnimmt. ■

Def. **Ausgabeeinheit**:

Eine Funktionseinheit innerhalb eines digitalen Rechensystems, mit der das System Daten, z.B. Rechenergebnisse, nach außen hin abgibt. ■

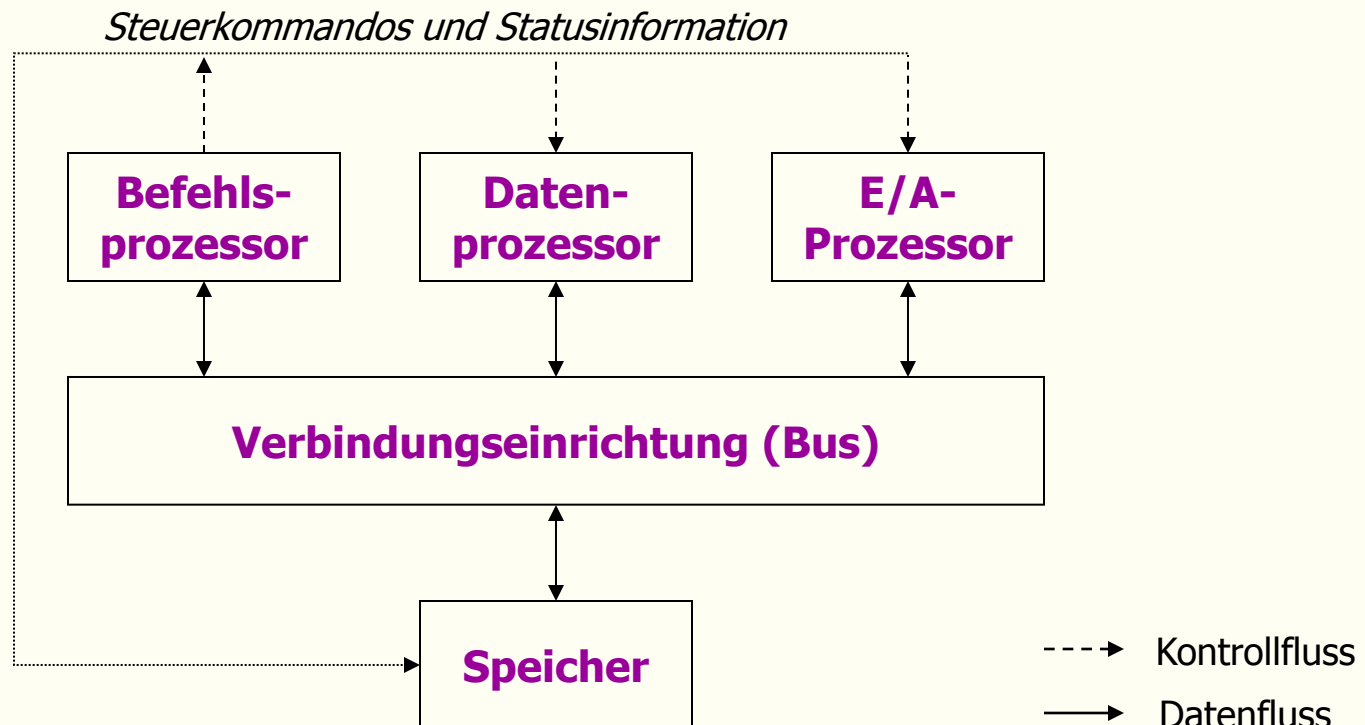
Hauptkomponenten von Rechensystemen (auf sog. "PMS-Processor-Memory-Switch"-Ebene)



nota bene: Wir sprechen auch von *Hardware-Betriebsmitteln*

Die Hauptkomponenten des von Neumann-Rechners und ihre Interaktionen

Zusammenspiel der Hauptkomponenten im **von Neumann-Rechner** (auch: "von Neumann-Maschine", "Princeton Computer" [Giloj 93])



→ vgl. *Burks, Goldstine, von Neumann* : "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument". Report to the U.S. Army Ordnance Dept., 1946

Wesentliche Eigenschaften der von Neumann-Maschine

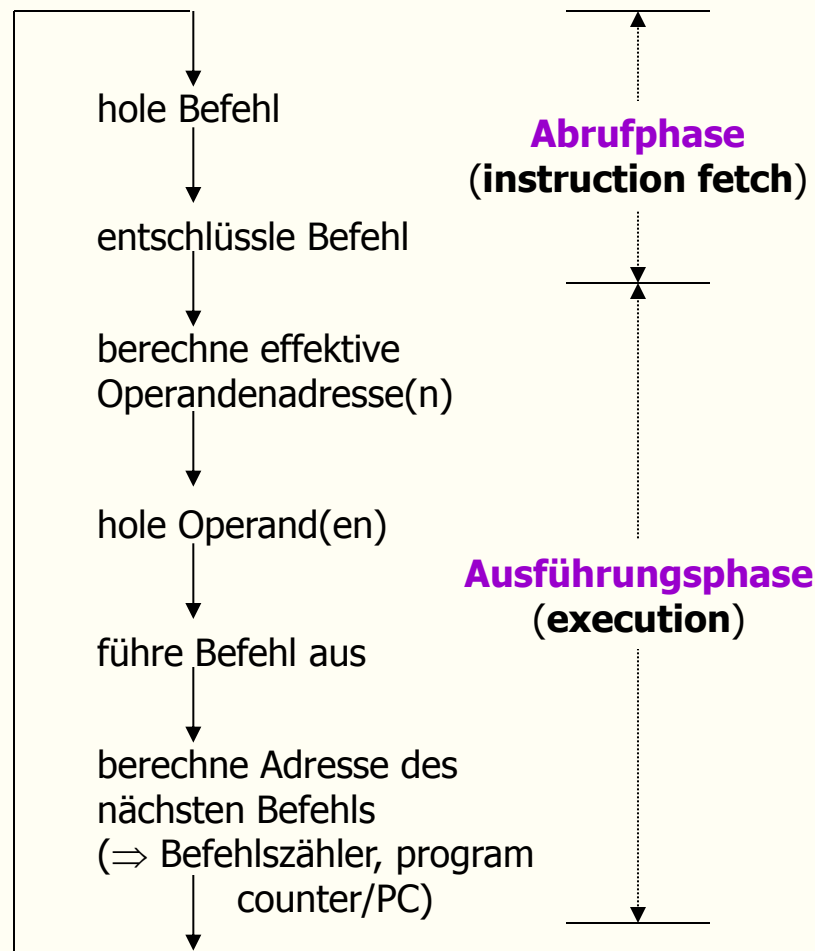
- Speicher für Programm und Daten / Operanden sind Einheit (→ Schreib-/Lesespeicher benötigt; Anweisungen und Operanden bzgl. Speicherung gleichbehandelt ⇒ fehleranfälliges Konzept, u.a. zustandsabhängige Interpretation eines gelesenen Bitmusters)
- Spez. Speicheraufbau : Einteilung in fortlaufend nummerierte Zellen (Nummern = Adressen)
- Nur Maschinenprogramme (Folgen elementarer Anweisungen) verarbeitet
- Befehlsaufbau : Operator + ggf. Adresse(n) des bzw. der Operanden
- Abweichungen von vorgegebener Befehlsreihenfolge durch Sprungbefehle (bedingt, unbedingt) → Sprungziel anstelle von Operandenangabe
- Verwendung von Binärzeichen/-signalen zur Darstellung aller Daten

Wesentliche Erweiterungen des von Neumann-Konzepts (in ca. 60 Jahren !)

- Software unterstützt wesentlich andere Benutzerschnittstellen und Betriebsformen
- Gleichzeitige Prozesse in Funktionseinheiten (Füllung > 1)
- Mikroprogrammierung (z.B. von Rechenwerk, E/A-Werk)
- Vielfältige Strukturierung von Befehlen (z.B. ≥ 2 Adressen möglich); versch. Befehlstypen mit unterschiedl. Formaten
[jedoch : gegenläufige Tendenz \rightarrow RISC, **r**educed **i**nstruction **s**et **c**omputers]
- Adressierung über Folge von Transformationen (z.B. Adressierung relativ zu Programmbeginn)
- Speicherhierarchie \rightarrow Entschärfung des „von Neumann Bottleneck“ (Flaschenhals)
- Auch externe Signale (Unterbrechungen) durch Leitwerk berücksichtigt
- Mehrfache Anordnung der von Neumann'schen Funktionseinheiten
 \Rightarrow Feldrechner, Multiprozessorsysteme, Rechnernetze (s.u.)

Zur Abarbeitung von Maschinenbefehlen

Der *(Maschinen-)Befehlszyklus*:



evtl.:

- Berücksichtigung von Unterbrechungswünschen
- ggf. triviale Ausführungsphasen (z.B. bei Sprungbefehlen)
- kein Holen von Operanden
- Holen von ≥ 2 Operanden

nota bene: hier kontrollflussorientierte Verarbeitung (andersartige Abläufe z.B. bei datenflussorientierter Verarbeitung !)

n-Adressmaschinen

Befehlsformate :

Op_Code: Befehlscode = Art/Typ des Befehls, z.B. **ADD(iere)**

➤ **n-Adressmaschine :**

Op_Code	Operand_1	Operand_2	...	Operand_n
---------	-----------	-----------	-----	-----------

➤ **3-Adressmaschine :**

Op_Code	Operand_1	Operand_2	Result
---------	-----------	-----------	--------

z.B.: **Result := ADD (Operand_1, Operand_2)**

➤ **2-Adressmaschine :**

Op_Code	Operand_1	Operand_2
---------	-----------	-----------

z.B.: **Operand_2 := ADD (Operand_1, Operand_2);** sog. Überdeckung (von Operand_2)

➤ **1-Adressmaschine :**

Op_Code	Operand 1
---------	-----------

z.B.: **ACCUM := ADD (Operand_1, ACCUM);** sog. Überdeckung (von ACCUM als spez. Register)

➤ **0-Adressmaschine :**

Op_Code	PUSH	Hsp_Adr	POP	Hsp_Adr
---------	------	---------	-----	---------

(„*Stack-Rechner*“)

z.B.: **Stack_1 := ADD (Stack_1, Stack_2);** Stack_i = i-ter Stack-Eintrag (von oben)

PUSH (Hsp_Adr) bringt Inhalt (Hsp_Adr) auf Stack;

POP (Hsp_Adr) bringt Inhalt des obersten Stack-Eintrags in den Hsp nach Adr.: Hsp_Adr.

Beispiel für Arbeitsweise einer 2-Adressmaschine

Aufgabe: Berechnung von

$$Z := (A - B * C) / (B + D / E)$$

Annahme : Überdeckung des 2. Operanden; H1 / H2 : Hilfsspeicherzellen;
>X< bezeichne Adresse der Zelle, die X beinhaltet

	Befehlsfolge		Wirkung
(1) →	>B<	>H1<	H1 := B
(2) *	>C<	>H1<	H1 := C * H1
(3) -	>A<	>H1<	H1 := A - H1
(4) →	>E<	>H2<	H2 := E
(5) /	>D<	>H2<	H2 := D / H2
(6) +	>B<	>H2<	H2 := B + H2
(7) /	>H1<	>H2<	H2 := H1 / H2
(8) →	>H2<	>Z<	Z := H2

Ergo: **Gesamtaufwand** für Lösung der Aufgabe:

8 Befehle mit

13 Leseaufträgen an Hsp./Cache

(jedoch nur **6 Leseaufträge**, sofern H1 / H2 : Register)

Beispiel für Arbeitsweise einer 1-Adressmaschine

Aufgabe: Berechnung von

$$Z := (A - B * C) / (B + D / E)$$

Annahme : Überdeckung des Akkumulator-Inhaltes; AC: Akkumulator;
>X< bezeichne Adresse der Zelle, die X beinhaltet

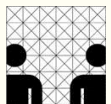
Befehlsfolge		Wirkung
(1) lade	>D<	AC := D
(2) /	>E<	AC := AC / E
(3) +	>B<	AC := AC + B
(4) speichere	>H1<	H1 := AC
(5) lade	>B<	AC := B
(6) *	>C<	AC := AC * C
(7) speichere	>H2<	H2 := AC
(8) lade	>A<	AC := A
(9) -	>H2<	AC := AC - H2
(10) /	>H1<	AC := AC / H1
(11) speichere	>Z<	Z := AC

} evtl. ersetzbar durch
Befehl mit Wirkung :
AC := A - AC

Ergo: **Gesamtaufwand** für Lösung der Aufgabe:

11 Befehle mit

8 Leseaufträgen an Hsp./Cache



Beispiel für Arbeitsweise einer 0-Adressmaschine

Aufgabe: Berechnung von

$$Z := (A - B * C) / (B + D / E)$$

Ausgangspunkt:

sog. Postfix-Notation (s.u.)

A B C * - B D E / + /

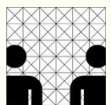
Annahme : In 0-Adressmaschine (auch: Keller-Rechner bzw. "Stack Computer") exist. Befehle **PUSH**, **POP** (für Transfers Hsp ↔ Stack), **ADD**, **SUB**, **MUL**, **DIV**, ... (arithmet. Befehle)

Befehlsfolge

Kellerinhalt

(1)	PUSH	>A<	A			
(2)	PUSH	>B<	A	B		
(3)	PUSH	>C<	A	B	C	
(4)	MUL		A	B*C		
(5)	SUB		A-B*C			
(6)	PUSH	>B<	A-B*C	B		
(7)	PUSH	>D<	A-B*C	B	D	
(8)	PUSH	>E<	A-B*C	B	D	E
(9)	DIV		A-B*C	B	D/E	
(10)	ADD		A-B*C	(B+D/E)		
(11)	DIV		(A-B*C) / (B+D/E)			
(12)	POP	>Z<				

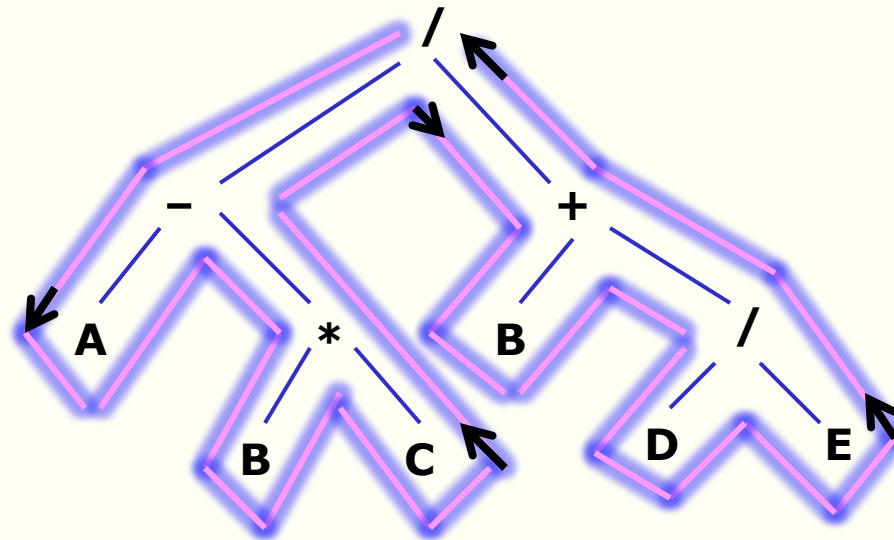
Ergo: **Gesamtaufwand** für Lösung der Aufgabe: **12 Befehle** mit **6 Leseaufträgen** an Hsp./Cache



Herleiten der Postfix-Notation zu gegebenem Ausdruck

Aufgabe: Berechnung von

$$Z := (A - B * C) / (B + D / E)$$



Resultat:

Ausdruck in Postfix-Notation

A B C * - B D E / + /

Zur Adressierung von Operanden

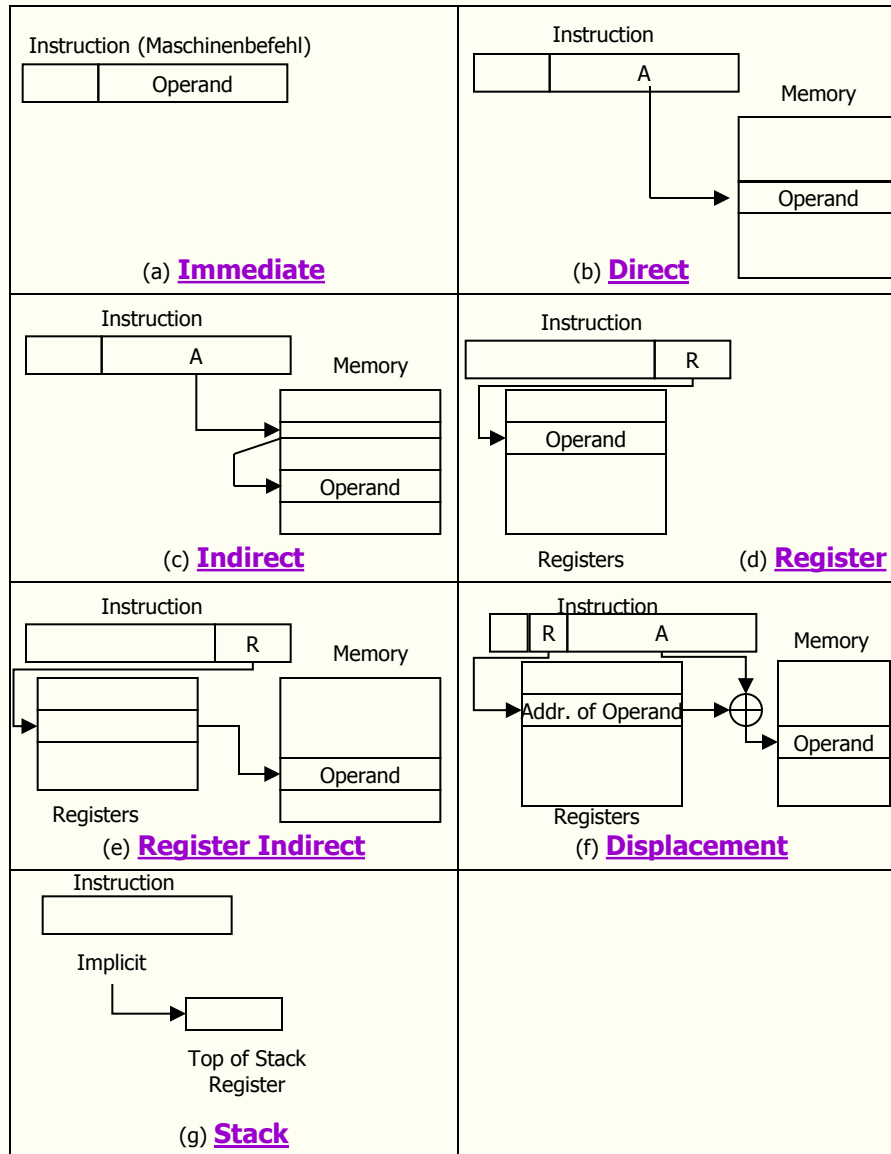


Fig.: **Addressing Modes**

[Sta 96] – W. Stallings: *Computer Organisation and Architecture* (Prentice-Hall, 1996)

Abstraktionsebenen zur Diskussion von Rechnerarchitekturkonzepten

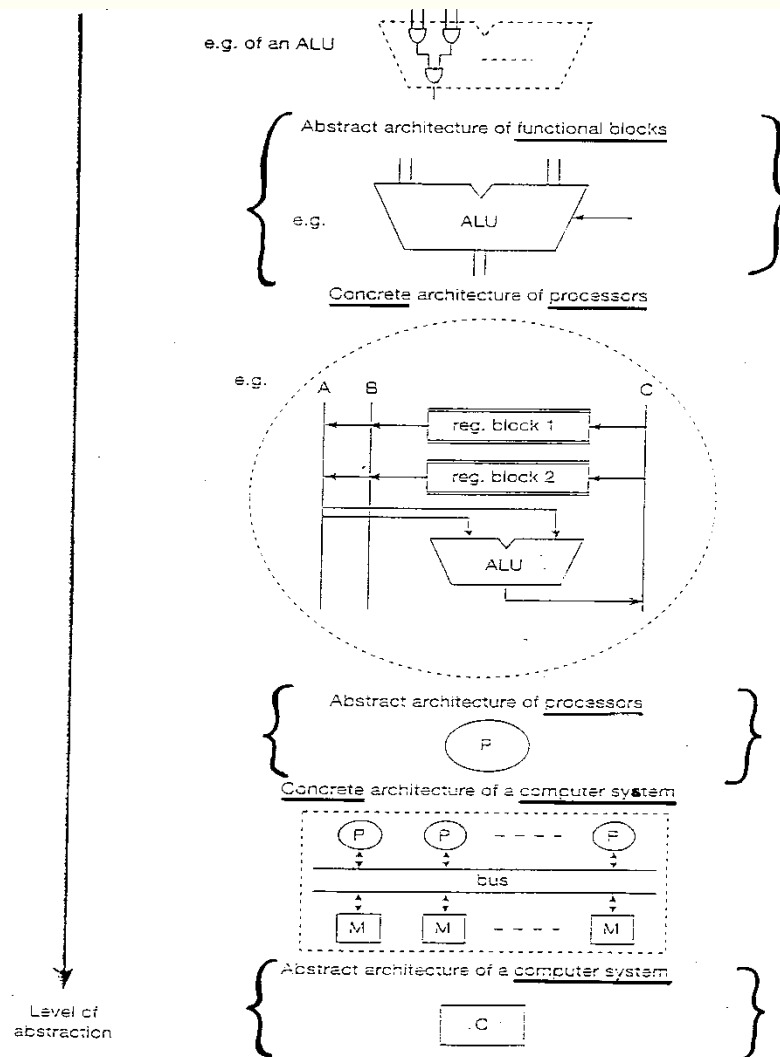


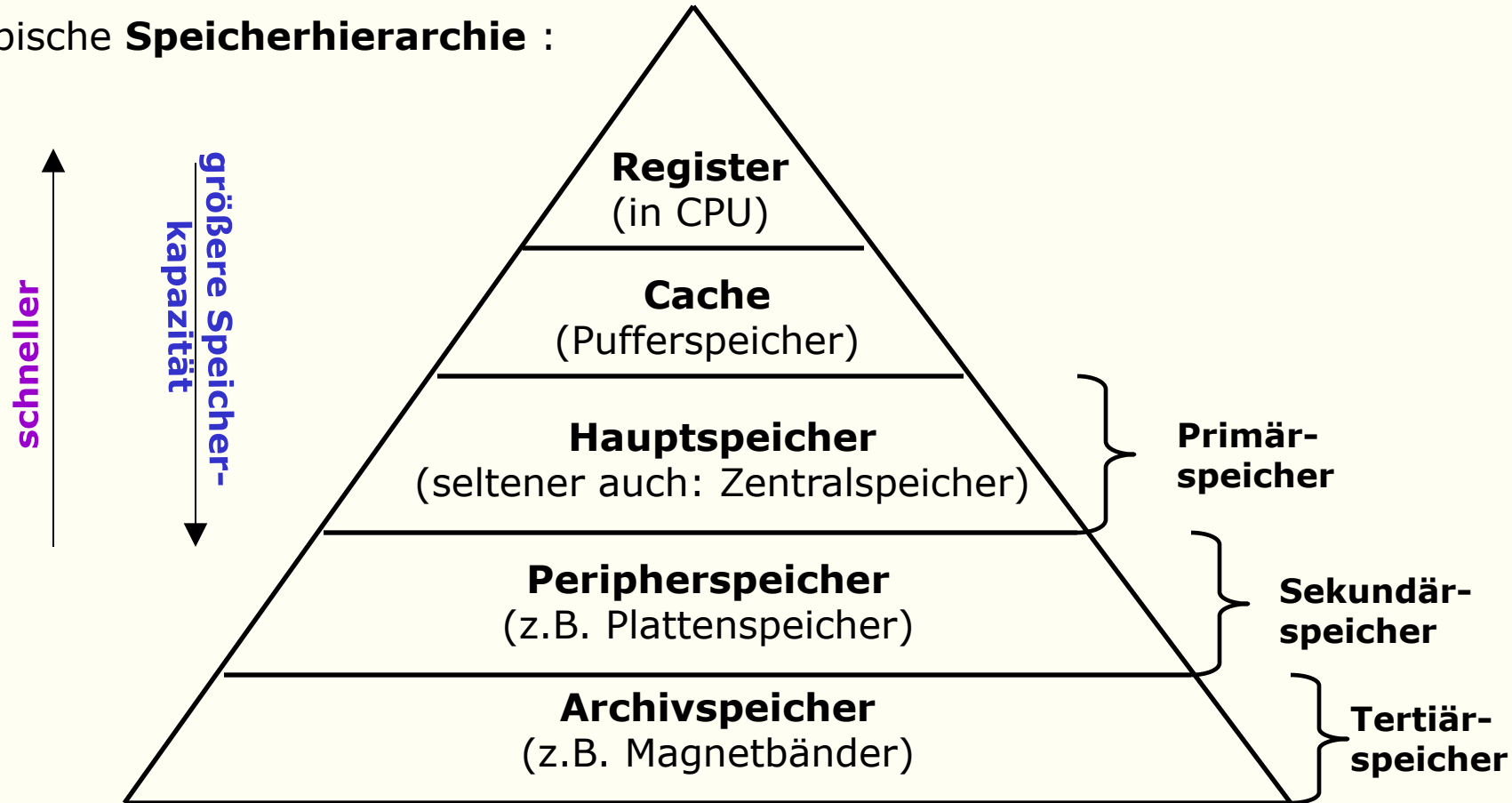
Figure 2.10 Architecture concepts at increasing levels of abstraction.

Speicherhierarchie

Gründe für Speicherhierarchien:

- schneller Speicher ist relativ teuer
- großer Speicher ist relativ langsam (große Zugriffszeiten)

⇒ Typische **Speicherhierarchie** :



A1.2 Betriebssysteme : Historische Entwicklung und einige Grundkonzepte

Systemsoftware als “Bindeglied” zwischen Programmen und Hardware

Das Doppelgesicht der Systemsoftware:

PROGRAMM

**Barriere
der Systemsoftware**

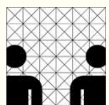
HARDWARE

Negativ:

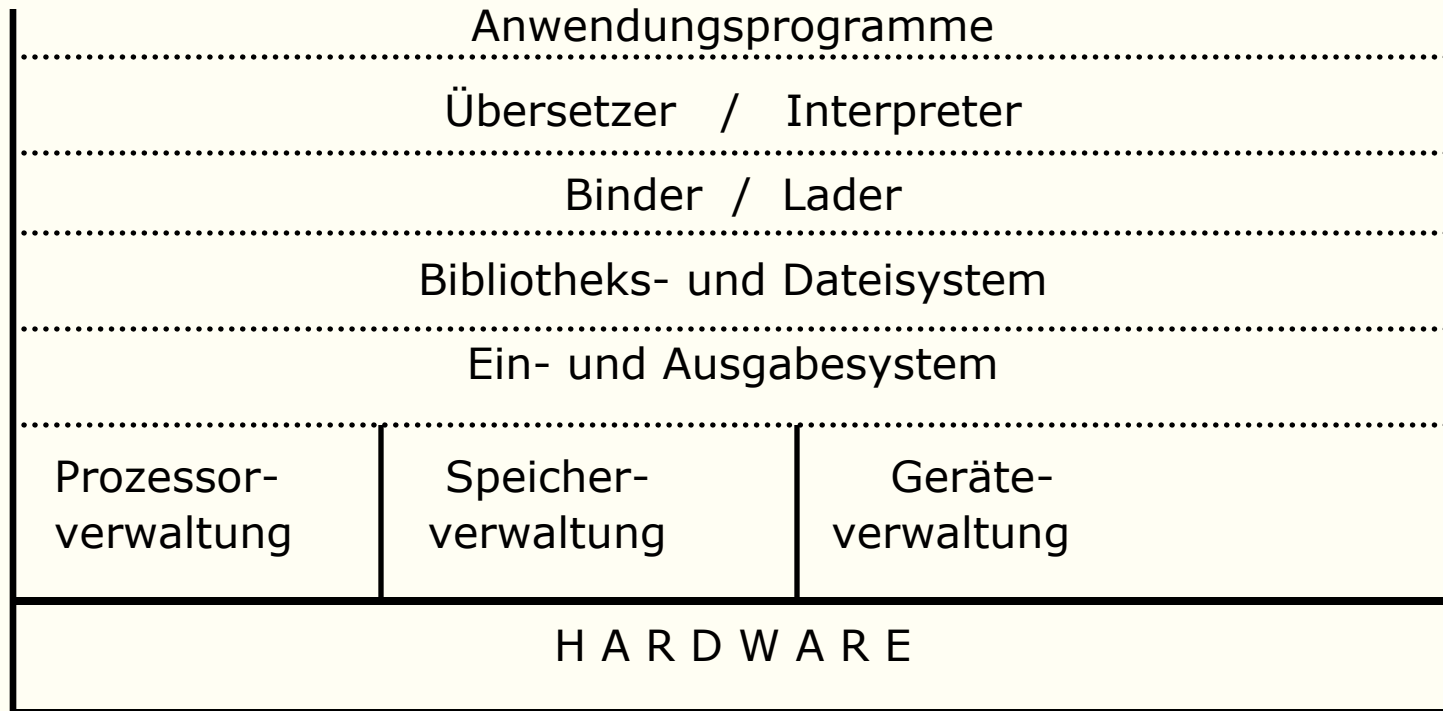
- Die Systemsoftware verbraucht wertvolle Betriebsmittel.
- Die Systemsoftware zwingt den Programmierer zur Beachtung eines Wustes von Vorschriften.

Positiv:

- Entlastung des Programmierers von den Idiosynkrasien der technischen Geräte-realisation.
- Unterstützung des Programmierers bei der Erstellung und Verwaltung von Programmen.



Grobes Modell eines Rechensystems



Fragen:

- (i) Welche der obigen Tätigkeiten gehören in ein Betriebssystem?
- (ii) Wohin legt man die Sicherheitsschicht?
- (iii) Nach welchen allgemeinen Konstruktionsprinzipien werden Systeme erstellt?

Aufgaben eines Betriebssystems

Die zwei **Grundaufgaben** eines Betriebssystems (BS):

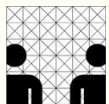
- I. Ein Betriebssystem verbirgt die Hardwaredetails und stellt dem Nutzer eine "angenehme" Arbeitsumgebung zur Verfügung. Dies führt zur Bildung plattformunabhängiger Betriebssysteme, z.B. Solaris, Windows NT, Windows Vista, OS400.
→ **BS als „virtuelle Maschine“**
- II. Ein Betriebssystem verwaltet Betriebsmittel im Auftrag von Nutzern.
→ **BS als „Betriebsmittelverwalter“**

Eine Listung der **Einzelaufgaben** eines Betriebssystems führt zu:

- Buchhaltung,
- Verwaltung von Betriebsmitteln,
- Erhöhung der Benutzbarkeit,
- Erhöhung der Zuverlässigkeit,
- Minderung der Maschinenabhängigkeit,
- Automatischer Operateur,
- Erleichterung menschlicher Kommunikation.

Einige bekannte Betriebssysteme

1955: GM OS (Das erste Betriebssystem ?)
1961: CTSS
1964: OS/360 (Ankündigung, erste Einführung 1967)
1965: Multics
1967: 'THE'- multiprogramming system
1969: TENEX
1969: Unix
1973: RT-11
1975: TOPS-20
1978: Apple DOS 3.1
1981: MS-DOS
1984: Macintosh OS
1993: Windows NT 3.1
1995: OS/390
2000: Windows 2000
2002: MAC OS X
2003: Fedora Core Linux
2012: Windows 8



Einige Daten zur Entwicklung von UNIX

- 1961 CTSS ("Compatible Timesharing System")
- 1965 Start von MULTICS
- 1969 Ken Thompson erstellt erste Version von Unix auf einer PDP-7.
- 1970 Ken Thompson überträgt Unix von der PDP-7 auf eine PDP-11, die Größe des Systems beträgt etwa 12.000 Byte.
- 1973 Unix wird weitgehend in C neu programmiert, etwa 10.000 Codezeilen in C und etwa 1000 Codezeilen in Assembler.
- 1976 Unix Version 6
- 1978 BSD-Unix; Unix Version 7
- 1983 Unix System V
- 1985 System V Interface Definition
- 1991 Solaris 2, Linux
- 1998 Sun Solaris 7
- 2001 Linux 2.4
- 2011 Linux Mint 11 (alias „Katya“): Mitte 2012 beliebteste Linux-Distribution in BRD vor Ubuntu, siehe <http://www.linux-aktuell.info/>

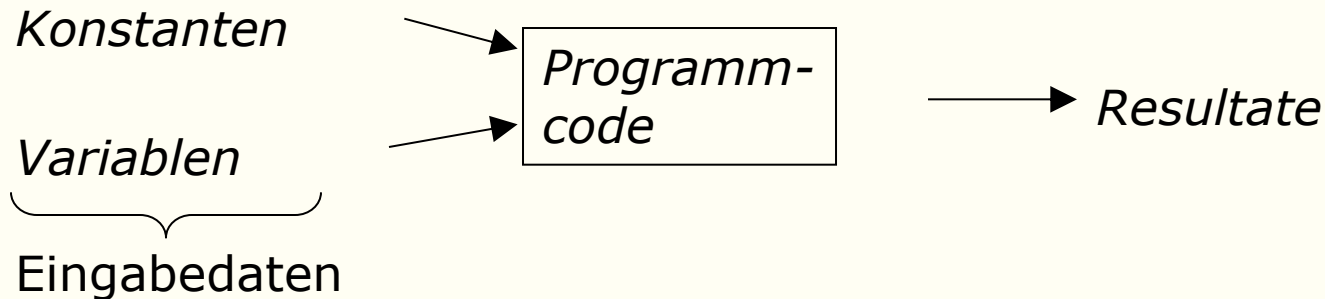
The Operating System Zoo

- *Mainframe* operating systems
- *Server* operating systems
- *Multiprocessor* operating systems
- *Personal computer* operating systems
- *Real-time* operating systems
- *Embedded* operating systems
- *Mobile* operating systems
- *Smart card* operating systems


(in Anlehnung an [Tan08])


Prozesse

➤ Programmablauf:



➤ Definitionen "Prozess":

Def. I: (Rechen-)Prozess \cong Ablauf eines Programms, sofern dieser durch das Betriebssystem verwaltet wird. 

Def. II: Prozess \cong Folge von Verarbeitungsschritten, deren erster begonnen, deren letzter aber noch nicht abgeschlossen ist. 

Beispiele für Verarbeitungsschritt: „Rechen-“, E/A-Operation (Betriebssystemaufruf), etc

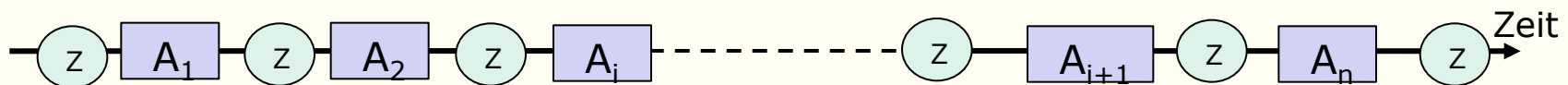
Prozesse

Bemerkungen:

- Prozess benötigt Prozessor zur Ausführung
- Jedem Prozess ist Prozessadressraum zugeordnet, u.a. mit
 - Programmcode
 - Konstanten
 - prozess-spezifischen Variablen

nota bene: Momentanzustand Z der Prozessausführung i.d.R. in dedizierten Betriebssystem-Tabellen, als sog. Prozesstabelle geführt (beinhaltet u.a. aktuelle Registerinhalte für Prozess, insbesondere Befehlszähler).


- **sequentieller Prozess:** Prozess mit *linearer* Folge von *Verarbeitungsschritten* (auch *Aktivitäten* A , s.u., genannt)

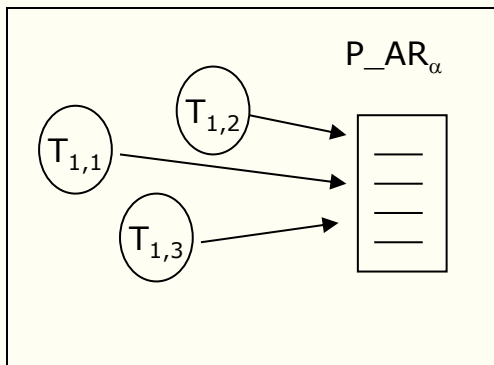


Leichtgewichtsprozesse/ "Threads"

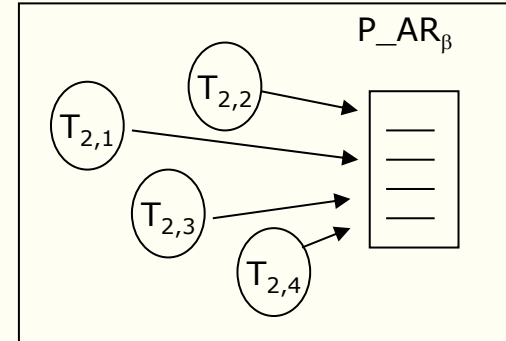
Def. „**Thread**“ (auch „*Leichtgewichtsprozess*“, engl. „*light weight process*“):

Programmabläufe, die

- durch das Betriebssystem verwaltet werden,
- einen gemeinsamen Prozessadressraum verwenden. 



Prozess 1
(mit Threads $T_{1,i}$)



Prozess 2
(mit Threads $T_{2,i}$)

P_AR: Prozess-Adressraum

Gründe für „Threads“

- effizienter Zugriff auf Daten des gemeinsamen Prozess-Adressraumes (P_AR)
- deutlich aufwandsärmere Erzeugung von Threads als von Prozessen
→ Threads auch für kurzfristige Aufgaben geeignet
- bei Prozesswechsel (CPU-Vergabe an neuen Prozess) ist im allg. P_AR nicht Hauptspeicher (Hsp)-resident
→ Zeitaufwand (z.B. für „Paging“, siehe Kap. A4);

bei Threadwechsel (zwischen Threads eines gemeinsamen Prozesses) befindet sich hingegen i.d.R. zumindest ein Teil des benötigten P_AR im Hsp.

Benutzerprozesse vs. Systemprozesse

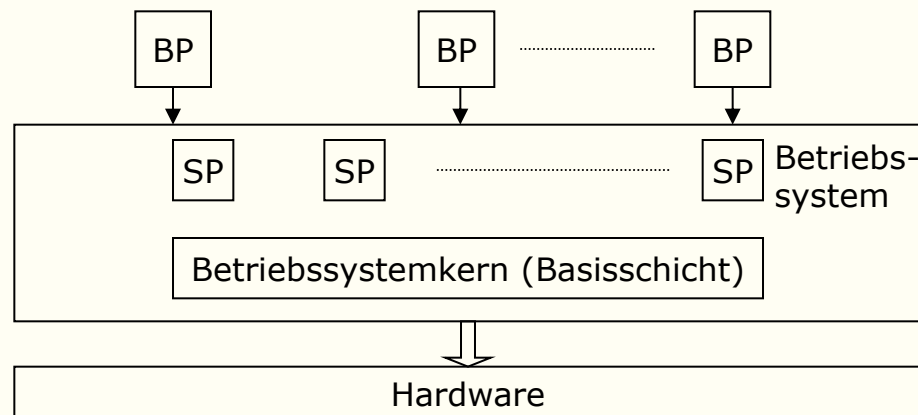
Def.

- **Benutzerprozesse** (BP): Prozesse, die Aufträge von Benutzern abwickeln (insbes. aus der Ausführung benutzerspezifischer Programme oder von Anwendungsprogrammen resultierend). ■
- **Systemprozesse** (SP): Prozesse, die ausgewählte Dienstleistungen des Betriebssystems (BS) erbringen. ■

Merkmale von Systemprozessen:

- häufig im Besitz spezieller Rechte
- im Systemmodus lauffähig
- sind wesentliche Betriebssystemkomponenten (neben dem **Betriebssystemkern** \cong nicht als Prozesse realisierte BS-Komponenten)

⇒ Prinzipieller Aufbau eines Betriebssystems:



Dateien

Datei:

längerfristig zu speichernde geordnete Datenmenge bestehend aus logisch zusammenhängender Menge von (Daten-)Sätzen. ■

... oder *allgemeiner*: Einheit der Speicherung beliebiger Daten auf einem persistenten (dauerhaften) Speicher [NeS98]. ■

Eigenschaften von Dateien:

- i.d.R. benutzerdefiniert
- sehr unterschiedliche Arten von Inhalten (z.B. Programmdateien, Eingabedaten, Festbilder, Videosequenzen, u.v.a.m.)
- Abstraktion von der physikalischen Speicherung, Konzept zur Realisierung von Geräte-/Speicherunabhängigkeit
- Organisation von Dateien in *Katalogen* bzw. *Verzeichnissen* (engl.: *directories*) → u.a. Gruppierung von Dateien in Verzeichnissen und Zugriff auf Dateien über Verzeichnisse
- benutzerspezifische Zugriffsrechte

Dateiverwaltung als wesentliche Aufgabe von Betriebssystemen

⇒ zu Details vgl. Kapitel A5. (und A4.)

Kommandointerpretierer/„Shell“

Benutzerschnittstellen von Betriebssystemen:

- graphische Oberfläche (GUI), oder
- Kommandoschnittstelle

Bei Kommandoschnittstelle kann ein Kommandointerpretierer (auch: „Shell“ bei Betriebssystemen der UNIX-Familie)

- beim Anmelden eines Benutzers gestartet werden
- auf Eingabe eines Kommandos warten
- die kommandospezifischen Betriebssystemoperationen initiieren, z.B. mittels geeigneter Betriebssystemaufrufe.

Typische Klassen von Betriebssystemaufrufen:

- Prozessverwaltung
- Signale
- Datei- und Geräteverwaltung
- Katalog- und Dateisystemverwaltung
- Schutzmechanismen
- Zeitverwaltung

Beispiele für Betriebssystemaufrufe

Aufrufklasse	Beispiele
Prozess- verwaltung	<ul style="list-style-type: none">• Ersetzen des Prozessadressraums• Beendigung der Prozessausführung mit Statusrückmeldung
Signale	<ul style="list-style-type: none">• Senden eines Signals an einen Prozess• Suspendierung des Anrufers bis nächstes Signal eintrifft
Datei- verwaltung	<ul style="list-style-type: none">• Erzeugen einer Datei• Schließen einer geöffneten Datei• Lesen von Daten aus einer Datei in einen Puffer
Katalog- und Dateisystem- verwaltung	<ul style="list-style-type: none">• Entfernen eines Katalogeintrags• Wechsel des aktuellen Arbeitskatalogs
Schutz- mechanismen	<ul style="list-style-type: none">• Veränderung der (Zugriffs-)Schutz-Bits einer Datei• Wechsel des Dateieigentümers (Benutzer und/oder Gruppe)
Zeitverwaltung	<ul style="list-style-type: none">• Setzen der Zeit des letzten Zugriffs auf eine Datei• Lesen der angefallenen Benutzer- bzw. Systemzeiten

Sicht I auf Betriebssysteme:

Betriebssystem als Betriebsmittelverwalter

Aufgaben des Betriebssystems bei Sicht I:

- Ermittlung des Betriebsmittel (BM)-Bedarfs von Prozessen → evtl. Problem: Bedarf a priori nicht präzise bekannt
- Allokation der BM zu Prozessen (auf Anforderung hin) → u.a. Fairness und Effizienz bei BM-Vergabe
- Ermittlung und ggf. Abrechnung der beanspruchten BM („accounting“)
- Behandlung von Problemen und Konflikten bei Inanspruchnahme von BM → u.a. Behandlung von Verklemmungs- und Engpass-Situationen.

Betriebsmittelverwaltung u.a. erschwert bei:

- großer Auswahl von um BM konkurrierenden Prozessen
- unpräzise spezifiziertem BM-Bedarf
- hoher Auslastung eines BM (zahlreiche zeitlich benachbarte Zugriffe und/oder geringe „Kapazität“ des BM, wie Speicher oder Verarbeitungsleistung)
- BM-Allokation in verteilten Systemen ! (→ vgl. die Kapitel zu Rechnernetzen)

Sicht II auf Betriebssysteme: Betriebssystem als „virtuelle Maschine“

Aufgaben des Betriebssystems bei Sicht II:

- Details der zugrundeliegenden Rechnerhardware den Benutzern verbergen (z.B. Unsichtbarkeit der Speicherhierarchie, der angeschlossenen E/A-Geräte und Netze sowie der benutzten Peripheralspeicher, etc.)
- Details der gleichzeitigen Mehrfachbenutzung des Rechners und seiner Komponenten dem einzelnen Benutzer verbergen
- Realisierung von (weitestgehender) Hardwareunabhängigkeit für die zu erstellenden Programme



durch Abstraktion von Hardware-Details:

Nutzung des Rechners deutlich komfortabler (Betriebssystem übernimmt die Abbildung*) der benutzernahen Dienste – z.B. Zugriff auf Dateien – auf die elementaren, systemnahen Basisfunktionen der Hardwarekomponenten).

*) *nota bene*: Diese Abbildung erfolgt aus Komplexitätsgründen im allgemeinen nicht direkt, sondern durch Abbildung auf sukzessive grundlegendere Dienste
(→ vgl. Diensthierarchien in Betriebssystemen, s. Kapitel B2.)