Systemprogrammierung

Teil 1: Einführung

Prof. Dr. H. Drachenfels
Hochschule Konstanz

Version 3.0
26.7.2013

Systemsoftware versus Anwendungssoftware

Systemsoftware dient dem Betrieb von Rechnern

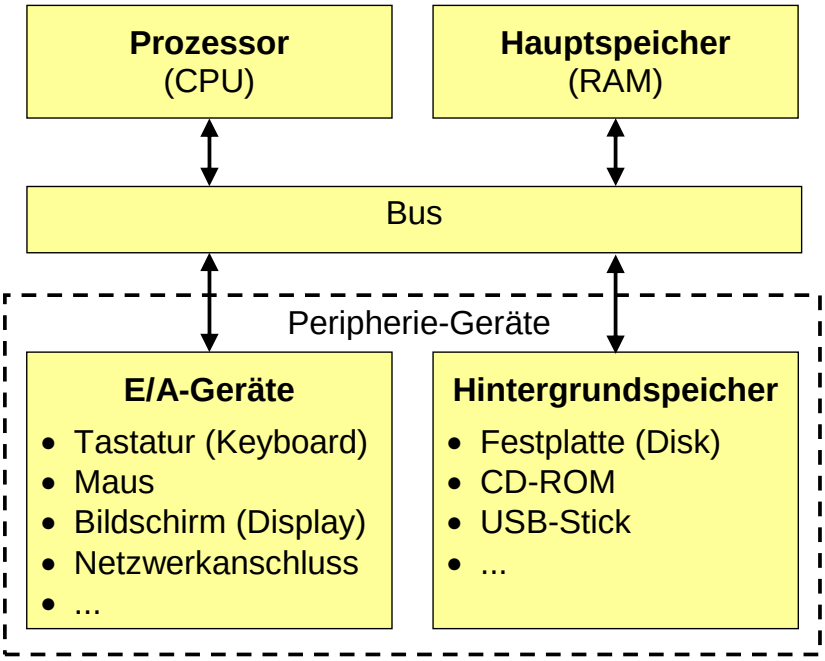
- Verwaltung der Hardware-Ressourcen und Steuerung der internen Abläufe
  Prozessor- und Speicherverwaltung, Kommunikation mit angeschlossenen Geräten usw.
- Bereitstellen einer komfortablen Ablaufumgebung für Anwendungssoftware
  Verbergen von speziellen Hardware-Eigenschaften usw.
- Beispiele:
  Betriebssysteme, Datenbanksysteme, Firmware, JVM (Java Virtual Machine), ...

Anwendungssoftware stellt Funktionalität für Endbenutzer eines Rechners bereit

- Verwaltung und Verarbeitung von Anwenderdaten
- Bereitstellen komfortabler Bedienoberflächen
- Beispiele:
  Browser, Textverarbeitung, Computerspiele, ...

Die Grenze zwischen Systemsoftware und Anwendungssoftware ist fließend.

Systemsoftware und Hardware



- Systemsoftware muss die Hardware-Ressourcen möglichst optimal nutzen

  sie muss insbesondere die Begrenztheit der Ressourcen beachten

- die Programmiersprache darf deshalb nicht zu stark von der Hardware abstrahieren

  das gilt insbesondere für die Ressource Speicher

Systemsoftware und Programmiersprachen

Ursprünglich wurde Systemsoftware vollständig in **Assemblersprachen** erstellt:

- eine Assemblersprache bietet lediglich lesbare Namen für Maschinenbefehle
- Software ist dadurch an den Befehlssatz einer Prozessorfamilie gekoppelt
- Programmierung mühsam und fehleranfällig

Heute wird Systemsoftware überwiegend in der Hochsprache **C** erstellt:

- übersichtliche Sprache mit sehr guter Werkzeugunterstützung
- in den 1970er-Jahren als Programmiersprache von Unix entstanden

  1978: Kernighan & Ritchie - *"The C Programming Language"*
  1989: ANSI-C, bis heute der am breitesten unterstützte Standard
  1990: C90, ISO-Standard, der bis auf Kleinigkeiten ANSI-C entspricht
  1995: C95, kleinere Ergänzungen
  1999: C99, Angleichungen an C++ und einige Erweiterungen
  2011: C11, unter anderem bessere Unicode-Unterstützung, Bibliothekserweiterungen

Systemprogrammierung: Inhalt der Lehrveranstaltung

Einführung in die Sprache ANSI-C

- Sprachkonzepte: Datentypen, Anweisungen, Funktionen, Übersetzungseinheiten
- Standardbibliothek: Speicherverwaltung, Ein-/Ausgabe, Dateien, ...

Werkzeuge

- Compiler: **`gcc`**
- Debugger: **`ddd`**, **`valgrind`**
- Automatisierung der Programmerstellung: **`make`**

Programmorganisation

- statische und dynamische Bibliotheken
- ausführbare Dateien
- Archive
- Prozesse