

Lernzettel

Technische Grundlagen der Informatik

Universität: Technische Universität Berlin
Kurs/Modul: Technische Grundlagen der Informatik
Erstellungsdatum: September 19, 2025



Zielorientierte Lerninhalte, kostenlos!
Entdecke zugeschnittene Materialien für deine Kurse:

<https://study.AllWeCanLearn.com>

Technische Grundlagen der Informatik

Lernzettel: Technische Grundlagen der Informatik

(1) Grundidee der Rechnerarchitektur

Die technische Grundlage der Informatik umfasst das Verständnis der Arbeitsweise eines Computers durch die Kombination von Bausteinen wie CPU, Speicher und Ein-/Ausgabegeräten. Zentrale Konzepte sind das Von-Neumann-Modell, die Speicherhierarchie und der Datenpfad zwischen Rechenwerk, Steuerwerk und Speicher.

(2) Aufbau eines typischen Computers (Von-Neumann-Architektur)

Ein typischer Computer besteht aus:

- Rechenwerk (ALU) und Steuerwerk,
- Speicherhierarchie mit Register, Cache und Hauptspeicher (RAM),
- Ein-/Ausgabeeinheiten und Busse für den Datentransfer,
- Ein Programmspeicher, der Befehle und Daten enthält.

Die CPU führt Befehle Schritt für Schritt aus, wobei der Ablauf durch den Steuerfluss gesteuert wird. Die Speicherhierarchie dient dazu, die Zugriffszeiten zu harmonisieren: schnellere, teurere Speicherbausteine (Register, Cache) arbeiten enger, langsamere Speicherbausteine (RAM, Festplatte) weiter außen.

(3) Rechnerarchitektur – Rechenwerk, Speicher und Busse

Rechenwerk (ALU): führt arithmetische und logische Operationen aus.

Steuerwerk: koordiniert den Ablauf der Befehle und den Zugriff auf Speicher.

Speicherhierarchie: Register < L1/L2 Cache < RAM < SSD/HDD.

Busse: Transportwege, die Daten, Adressen und Steuerinformationen zwischen Bausteinen übertragen.

(4) Von-Neumann-Flaschenhals und Leistungskennzahlen

Der Flaschenhals entsteht durch die gemeinsame Nutzung von Speicher und Rechenwerk. Leistungskennzahlen:

$$\text{Durchsatz} = \frac{\text{Anzahl bearbeiteter Anweisungen}}{\Delta t}, \quad \text{Latenz} = \text{Zeit bis zur Fertigstellung eines Befehls.}$$

Typische Optimierungen: Cache-Hierarchien, pipelining, Parallelität, Speicherkonsistenz-Modelle.

(5) Betriebssysteme – Aufgaben und Konzepte

Aufgaben des Betriebssystems:

- Ressourcenverwaltung (CPU, Speicher, I/O),
- Abstraktion von Hardware durch Systemaufrufe und Treiber,
- Prozessor- und Speichermanagement (Prozesse, Threads, Adressräume),

- Scheduling, Interrupt-Handling und Fehlerbehandlung.

(6) Prozesse, Threads und Scheduling

Prozess ist eine Ausführungseinheit mit eigenem Adressraum; **Thread** teilt sich Adressraum, ermöglicht leichtere Parallelität.

Scheduling bestimmt, welcher Prozess/Thread die CPU erhält. Typische Algorithmen:

- First-Come-First-Served (FCFS),
- Round-Robin (RR),
- Prioritätsbasierte Algorithmen.

Kontextwechsel: Wechsel von einem Prozess/Thread zu einem anderen (Kosten beachten).

(7) Speicherverwaltung und Virtuelle Speichertechnik

Virtueller Speicher verwendet Adressräume, die von der physischen RAM-Größe unabhängig sind.

Wichtige Konzepte:

- Paging: Speicher wird in Seiten und Rahmen zugeordnet,
- Seitentabellen, TLB (Translation Lookaside Buffer) als Cache der Seitentabellen,
- Speicherschutz und Privilegien (Benutzermodus vs. Kernelmodus).

(8) Rechnernetze und verteilte Systeme – Grundlagen

Schichtenmodell: OSI 7-Schichten oder TCP/IP-Architektur.

Wichtige Konzepte:

- Adressierung, Routing und Paketvermittlung,
- Grundprotokolle (z. B. TCP, UDP, IP),
- Fehlerbehandlung, Fluss- und Staukontrolle,
- Grundlagen verteilter Systeme: Transparenz, Replikation, Konsistenz, Fehlertoleranz.

(9) Sicherheit, Zuverlässigkeit und Performance-Grundlagen

Grundbegriffe: Schutzmechanismen, Zugriffskontrollen, Isolation, Fehlertoleranz, Wartbarkeit.

Leistungsaspekte berücksichtigen: Skalierbarkeit von CPU, Speicher und Netzwerk; Parallelität und Verteilte Systeme benötigen Synchronisation und Konsistenzmodelle.