

Lernzettel

Speicherverwaltung, Virtualisierung und
Speicherhierarchie: Paging, TLB, Cache,
Speicherallokation

Universität:	Technische Universität Berlin
Kurs/Modul:	Systemprogrammierung
Erstellungsdatum:	September 6, 2025



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

<https://study.AllWeCanLearn.com>

Systemprogrammierung

Lernzettel: Speicherverwaltung, Virtualisierung und Speicherhierarchie: Paging, TLB, Cache, Speicherallokation

(1) Grundbegriffe und Zielsetzung. Die Speicherhierarchie beschreibt Ebenen der Speicherzugriffe von schnell bis langsam (Register, L1/L2/L3 Cache, Hauptspeicher RAM, Festplatte/Swapping). Ziel ist, Zugriffszeiten zu minimieren und Kosten zu begrenzen. Caching sorgt durch Zwischenspeichern häufiger Zugriffe für geringere Latenzen.

(2) Paging und virtuelle Adressierung. Paging teilt den virtuellen Speicher in Seiten gleicher Größe und den physischen Speicher in Seitenrahmen (Frames). Die Übersetzung läuft über eine Seitentabelle.

$$\text{virtuelle Adresse } A = \text{VPN} \cdot \text{PageSize} + \text{Offset}, \quad 0 \leq \text{Offset} < \text{PageSize},$$

$$\text{physische Adresse } P = \text{PFN} \cdot \text{PageSize} + \text{Offset}.$$

- PageSize ist üblicherweise 4 KiB.
- VPN = virtuelle Seitennummer, PFN = physische Rahmennummer.

(3) Translation Lookaside Buffer (TLB). TLB ist ein schneller Cache der Seitentableneinträge. Übersetzung:

$$\text{TLB}[\text{VPN}] \rightarrow \text{PFN} \quad (\text{Hit}),$$

$$\text{bei Miss: PFN} = \text{PageTable}[\text{VPN}] \rightarrow \text{TLB aktualisieren}.$$

Die effektive Adressbildung erfolgt wie oben, jedoch mit dem im TLB gefundenen PFN.

(4) Speicherhierarchie und Cache. Cache-Typen: - direkt abgebildet (direct-mapped), - assoziativ (fully associative), - set-assoziativ.

Parameter eines Caches: - Größe C , - Blockgröße B , - Anzahl Sets S bzw. Linien.

Wichtige Konzepte: - Adressaufteilung in Tag, Index, Offset (Abhängigkeit von Mapping). - Durchschnittliche Zugriffszeit (AMAT):

$$\text{AMAT} = t_{\text{hit}} + (1 - h) t_{\text{miss}},$$

wobei h die Trefferquote ist.

(5) Speicherallokation (physischer Speicher). Ziel: effiziente Nutzung des RAM, Minimierung von Fragmentierung.

Typen von Allokation: - Freiliste (Free List), - Bitmasken (Bitmap), - Buddy-System (Buddy Allocator).

Strategien: - First-fit, Next-fit, Best-fit, Worst-fit (mit Vor- und Nachteilen je nach Anwendungsfall).

Im Betriebssystem erfolgt zusätzlich die Zuordnung von Seitenrahmen zu Prozessen (Paging) und ggf. Swapping/ Paging-out bei RAM-Engpässen.

(6) Page Replacement und Seitenfehler. Wenn kein freier Frame vorhanden, muss eine Seite ausgelagert werden.

Gängige Algorithmen: - LRU (Least Recently Used), - FIFO (First-In-First-Out), - Clock (eine praktische Approximation von LRU).

Seitenfehler-Typen: Hard Fault vs. Soft Fault. Die Wahl des Ersatzalgorithmus beeinflusst die Systemleistung stark.

(7) Virtualisierung der Speicherverwaltung. Gast-Adressraum wird in GVA (guest virtual address) übersetzt, dann in GPA (guest physical address) und schließlich in HPA (host physical address) durch MV bzw. EPT/Nested Page Tables.

Zentrale Aspekte: - Zwei-Stufen-Übersetzung ($GVA \rightarrow GPA \rightarrow HPA$), - TLB-Misses kostenintensiv, TLB-Shootdowns bei Kontextwechseln, - Nested Paging erhöht die Komplexität, aber Grenze der Übersetzung bleibt kritisch für Performance.

(8) Praktische Auswirkungen in C/Systemprogrammierung. - Speicherzugriffe formen lokale vs. globale Muster; Cache-Lokalität beachten. - Page-Größe und Alignment beeinflussen TLB- und Cache-Hits. - Exception Handling und Interrupt Handling im Zusammenhang mit Speicherzugriffen (z. B. Page Faults) beachten. - Sicherheitsrelevante Aspekte: Schutzmechanismen gegen Speicher-Daten-Lecks und -Diebstahl.

(9) Zusammenfassung. - Paging verbindet virtuellen und physischen Speicher; TLB beschleunigt Übersetzung. - Cache-Hierarchie reduziert Latenzen durch Ausnutzung von Temporal/Spatial Locality. - Speicherallokation und Virtualisierung bestimmen Ressourcennutzung, Leistung und Sicherheit moderner Systeme.