

Lernzettel

Speicherverwaltung im OS und Allokation:
Virtueller Speicher, Speicherallokation,
Fragmentierung

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: Speicherverwaltung im OS und Allokation: Virtueller Speicher, Speicher- allokation, Fragmentierung

(1) **Grundidee der Speicherverwaltung.** Die Speicherverwaltung dient dazu, Adressräume von Prozessen sicher, effizient und isoliert im Arbeitsspeicher abzulegen. Wichtige Aufgaben sind:

- Abbildung logischer Adressen auf physische Adressen (*Adressübersetzung*),
- Schutz und Isolation zwischen Prozessen,
- effiziente Nutzung des Speichers und Minimierung von Leerstellen,
- Unterstützung von Mehrprozessbetrieb und Overcommitment, falls sinnvoll.

(2) **Virtueller Speicher.** Beim virtuellen Speicher wird der für einen Prozess vorgesehene Adressraum nicht zwingend vollständig physisch im RAM gehalten. Stattdessen werden Adressübersetzung, Paginierung und ggf. Segmentierung verwendet.

- Adressaufbau: Eine virtuelle Adresse besteht üblicherweise aus Page-Number p und Offset d .
- Seitengröße $S = 2^n$ Bytes; virtuelle Adressräume haben $m + n$ Bits, mit $0 \leq p < 2^m$ und $0 \leq d < 2^n$.
- Übersetzung: $\text{PhysAddr} = \text{PFN} \cdot S + d$, wobei PFN der Physical-Frame-Nummer ist, die durch die Seitentabelle bestimmt wird.
- Seitentabelle (*page table*): Zu jeder virtuellen Seite p gehört der Eintrag, der den physischen Rahmen (PFN) oder ggf. einen Valid-Flag enthält.
- Translation Lookaside Buffer (*TLB*): Cache, der häufig genutzte Übersetzungen speichert.

$$\text{VirtAddr} = p \cdot S + d, \quad S = 2^n, \quad \text{PhysAddr} = \text{PFN} \cdot S + d$$

(3) **Speicherallokation – Überblick.** Speicherallokation behandelt, wie Prozesse Speicher beantragen und der physische Speicher zugeteilt wird. Grundtypen sind:

- Kontinuierliche Allokation (partitionierte Allokation): Prozesse erhalten zusammenhängende Blöcke; führt oft zu externer Fragmentation.
- Paging/virtueller Speicher: Logische Adressen werden in Seiten zerlegt; physischer Speicher wird in Seitenrahmen verwaltet; externe Fragmentation entfällt.
- Segmentierung (mit Paging kombiniert): logische Segmentgrößen werden separat verwaltet, kombiniert mit Paging.

(4) **Fragmentierung.** Fragmentierung beschreibt ungenutzte Speicherbereiche, die eine neue Zuweisung verhindern.

- Interne Fragmentation: Entsteht, wenn Zuweisungsgrößen größer sind als der tatsächlich genutzte Speicher (z. B. feste Partitionen).

- Externe Fragmentation: Freier Speicher ist in viele kleine Stücke zerstreut, sodass keine ausreichend große Lücke für eine neue Anforderung existiert.
- Maßnahmen zur Reduktion:
 - Paging reduziert externe Fragmentation, da Speicher in gleich große Seitenrahmen organisiert wird.
 - Kompaktion (Speicherverdichtung) verschiebt Blöcke, um größere freie Bereiche zu schaffen.
 - Verwendung von *slab*-Allokatoren, Buddy-Systeme, oder Segmentierung mit Paging.

(5) Formeln und Kennzahlen.

- Interne Fragmentation (bei fester Zuweisung):

$$\text{IF} = \sum_i (\text{Größe der Zuweisung}_i - \text{genutzter Speicher}_i)$$

- Externe Fragmentation (qualitativ; Anteil unbenutzbarer freier Speicherblöcke):

$$\text{EF} = \text{Summe der zu kleinen freien Blöcke}$$

- Fragmentierungsgrad:

$$\text{FG} = \frac{\text{Größe relevanter freier Speicherblöcke}}{\text{Gesamtfreier Speicher}}$$

- Seitenübersetzung (Beispiel): Für eine virtuelle Adresse V mit Seitenzahl p und Offset d ,

$$V = p \cdot S + d, \quad S = 2^n, \quad \text{PhysAddr} = \text{PFN}[p] \cdot S + d$$

(6) Ablaufbeispiel: Virtueller Speicher mit Paging.

- Prozessanforderung: Beantragt Adressraum $m + n$ Bits.
- Seitentabelle wird aufgebaut oder ins RAM/Cache geladen.
- Zugriff auf Adresse $V = p \cdot S + d$ prüft den TLB; bei TLB-Miss Hash-Tabelle oder Page Table Zugriff.
- Falls Seite nicht im RAM, Seitenfehler (*page fault*) und Laden der Seite von Festplatte in einen freien Rahmen erfolgt.

(7) Zusammenfassung. Der virtuelle Speicher ermöglicht Isolation, Schutz und effiziente Nutzung des RAMs. Paging reduziert externe Fragmentation, bringt jedoch Seitenfehlerkosten in Cycles. Unterschiedliche Allokationsstrategien beeinflussen Fragmentierung, Overhead und Zugriffsgeschwindigkeit. Verständnis von Seiten- und Segmentierungsstrukturen ist zentral für das Betriebssystemdesign.