



# Grundlagen digitaler Systeme

## Kapitel 16: Virtueller Speicher

Holger Karl

# Was bisher geschah

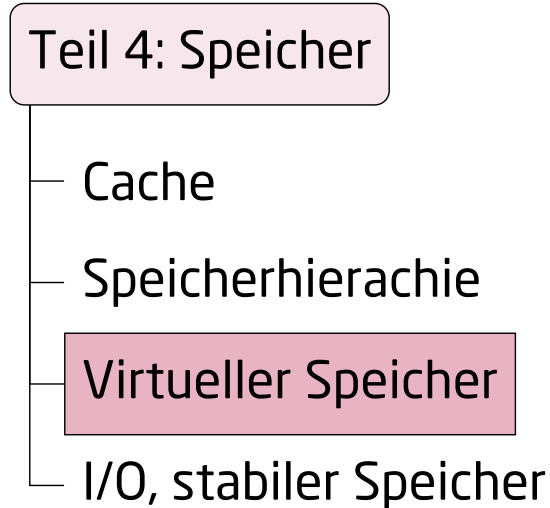
---

- Wir haben eine CPU mit Speicher und Cache ausgestattet
- Wir haben von einem Cache zu mehrstufigen Caches zu einer Speicherhierarchie erweitert
  - Die sich im Multiprozessor/-Core-Fall nach oben verzweigte
- Dabei war der Hauptspeicher (unausgesprochen) der “eigentlich” Ort des Wertes

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

# Plan für dieses Kapitel

- Was passiert, wenn der Hauptspeicher voll ist?
- Können wir Speicherhierarchie “nach unten” erweitern, in dem wir weitere Speicherarten hinzufügen?
- Kann eine Speicherhierarchie auch nach unten verzweigen? Konsequenzen?
- Hat das Auswirkungen auf Programmausführung?



**Abbildung 16.1:** Virtueller Speicher

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

- Gründe und Einsatzzwecke für virtuellen Speicher erläutern
- Wesentliche Mechanismen kennen; unterschiedliche Strategie einschätzen und Vor-/Nachteile abwägen
- Konsequenzen von Entwurfsentscheidungen auf System-Design, Programmausführung einschätzen
- Relevanz unterschiedlicher Parameterwerte bei strukturell ähnlichen Systemen für Entwurfsentscheidungen einschätzen; Entscheidungen daraus ableiten

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

# Inhaltsverzeichnis

---

## 1. Hintergrundspeicher

## 2. Virtueller Speicher

## 3. Programmausführung

## 4. Zusammenfassung

## 5. Material

## Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 5/66**

# Weitere Speichertypen hinzufügen?

---

- Speichertypen bis jetzt:
  - ☐ Register, direkt im Core
  - ☐ Cache, auf mehreren Ebenen, SRAM oder DRAM
  - ☐ Hauptspeicher, DRAM

## Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

# Weitere Speichertypen hinzufügen?

- Speichertypen bis jetzt:
  - ☐ Register, direkt im Core
  - ☐ Cache, auf mehreren Ebenen, SRAM oder DRAM
  - ☐ Hauptspeicher, DRAM
- Welche weiteren Speichertypen kommen in Frage?
  - ☐ Festplatten (magnetisch, rotierend, . . . )
  - ☐ Flash-Speicher
  - ☐ Sog. **Hintergrundspeicher**

## Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

# Hintergrundspeicher: Eigenschaften

- Erheblich langsamer als RAM
  - Sowohl Datenrate als auch Zugriffslatenz
  - Festplatte: Rotationslatenz
  - In Rate: SRAM > DRAM > Flash > Festplatte
  - In Latenz: SRAM < DRAM < Flash < Festplatte
- Viel billiger pro Einheit
  - In Geld: SRAM > DRAM > Flash > Festplatte
- Behält Inhalt auch ohne Stromzufuhr
- Adresse im Hintergrundspeicher: Technologieabhängig!
  - Z.B. Festplatte: Welcher Kopf, welche Spur, welcher Sektor

## Hintergrundspeicher

Virtueller Speicher

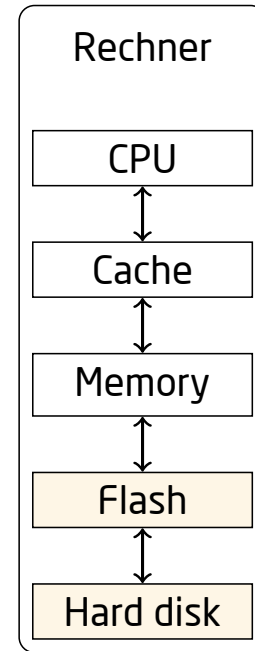
Programmausführung

Zusammenfassung

Material



- Einordnung: Unterhalb des Hauptspeichers
- Und dann: Hauptspeicher als Cache des Hintergrundspeichers auffassen?

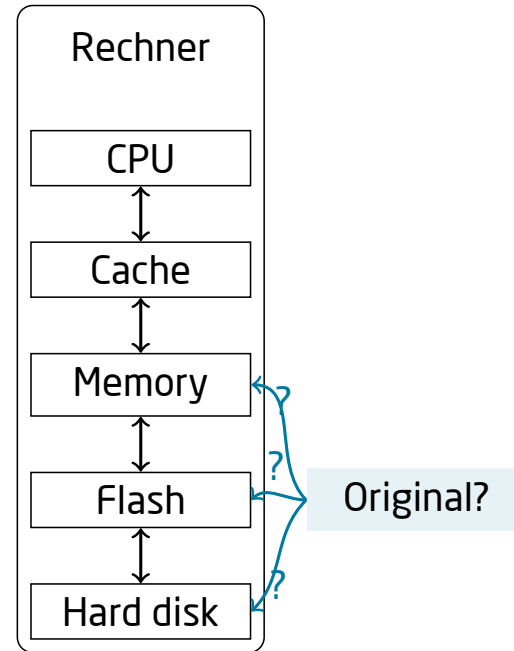


**Abbildung 16.2:** Erweiterung der Speicherhierarchie um Flash und Festplatten als Hintergrundspeicher

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

# Erweiterte Hierarchie: Original?

- Wo ist das "Original" eines Wertes?



**Abbildung 16.3:** Wo ist das Original eines Wertes?

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

# Erweiterte Hierarchie: Original?

- Wo ist das "Original" eines Wertes?
- Letztlich nicht relevant!
- Es geht nur darum, wie aktuell eine Kopie ist

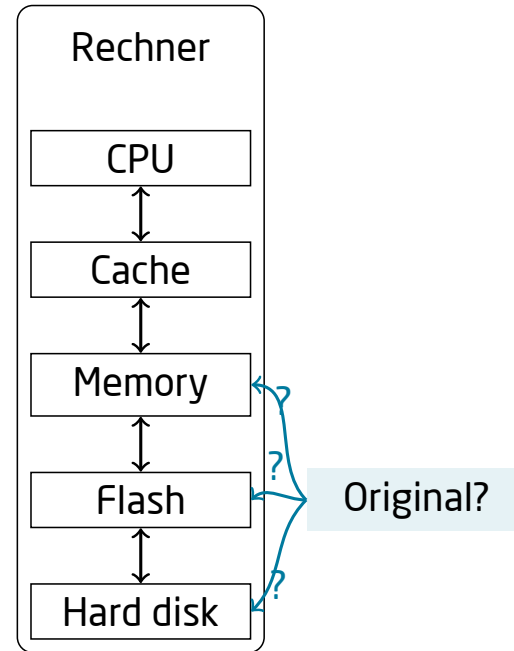


Abbildung 16.3: Wo ist das Original eines Wertes?

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 9/66**

# Übertragung der Cache-Konzepte?

---

- Können wir nun die bisherigen Cache-Konzepte direkt weiterbenutzen?
  - Caches mit Blöcken, Assoziativität des Caches, . . .

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 10/66**

# Übertragung der Cache-Konzepte?

---

- Können wir nun die bisherigen Cache-Konzepte direkt weiterbenutzen?
  - Caches mit Blöcken, Assoziativität des Caches, . . .
- Ja, im Prinzip schon

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 10/66**

# Übertragung der Cache-Konzepte?

- Können wir nun die bisherigen Cache-Konzepte direkt weiterbenutzen?
  - Caches mit Blöcken, Assoziativität des Caches, . . .
- Ja, im Prinzip schon
- Aber:
  - In Cache und Hauptspeicher konnten wir **identische Adressen** verwenden: letztlich laufende Nummer
    - Auch bei Adressumsetzung im vollassoziativen Cache
  - Das geht bei Hintergrundspeicher nicht ohne weiteres
    - Wir brauchen eine **Adressumsetzung**, um Adressen des Hauptspeichers entsprechende Teile des Hintergrundspeichers zuzuordnen - die mit anderen Adressformaten angesprochen werden müssen
    - Komplexer als Adressumsetzung eines vollassoziativen Caches!

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 10/66**

# Komplexe Adressumsetzung als Gelegenheit?

---

## Idee 16.1 (Komplexe Adressumsetzung)

- Wenn wir ohnehin eine komplexere Umsetzung brauchen, um Hintergrundspeicher zu nutzen,
- Dann können wir damit doch zusätzliche Funktionalität integrieren?

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material

## 1. Hintergrundspeicher

## 2. Virtueller Speicher

### 2.1 Virtuelle Adressen

### 2.2 Adressumsetzung

### 2.3 Seitentabelle

### 2.4 MMU und TLB

### 2.5 Seitenfehler

### 2.6 Verdrängung

### 2.7 Zusammenfassung

## 3. Programmausführung

## 4. Zusammenfassung

## 5. Material

## Hintergrundspeicher

### Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

## Programmausführung

## Zusammenfassung

## Material

## GDS 16: Virtueller Speicher

H. Karl, WS 22/23

**Folie 12/66**



# Zusätzliche Funktionalität: Größerer Adressraum

- Hintergrundspeicher hat mehr Platz
- Nutzbar machen, mit dem gleichen Instruktionssatz?
  - Laden, schreiben wie bisher, nur auf **mehr Adressen**?

## Definition 16.1 (Speichervirtualisierung)

Die Bereitstellung von **mehr Arbeitsspeicher** als tatsächlich vorhanden ist, **ohne** die Nutzungsweise zu ändern. Als Ressource zur tatsächlichen Datenspeicherung wird Hintergrundspeicher genutzt.

Man sagt, der Arbeitsspeicher wird **virtualisiert**. Speichermangel wird verborgen.

Hintergrundspeicher

**Virtueller Speicher**

Virtuelle Adressen  
Adressumsetzung  
Seitentabelle  
MMU und TLB  
Seitenfehler  
Verdrängung  
Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 13/66**

- Bisher haben wir **reale Adressen** benutzt: Jede darstellbare Adresse entsprach einer adressierbaren Einheit im Arbeitsspeicher
  - Byte, Wort, . . . - je nach Speicherorganisation, Breite des Datenbuses
  - Möglicherweise war der Arbeitsspeicher nicht voll ausgebaut, aber die Systemarchitektur, Adressbus, . . . sahen das vor
- Bitanzahl reale Adresse = Anzahl Leitung des Adressbusses

Hintergrundspeicher

Virtueller Speicher

**Virtuelle Adressen**

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 14/66**

# Virtuelle Adressen

---

- Um mehr Speicher anzusprechen brauchen wir mehr Bits
- Also: andere Sorte von Adressen

## Definition 16.2 (Virtuelle Adressen)

Virtuelle Adressen sind Adressen, die

- strukturell realen Adressen entsprechen
- aber idR mehr Bits enthalten
- in Instruktionen dennoch direkt genutzt werden können.

Sie müssen in reale Adressen umgesetzt werden, um tatsächliche Speicherinhalte anzusprechen.

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 15/66**

# Virtuelle Adressen

- Um mehr Speicher anzusprechen brauchen wir mehr Bits
- Also: andere Sorte von Adressen

## Definition 16.2 (Virtuelle Adressen)

Virtuelle Adressen sind Adressen, die

- strukturell realen Adressen entsprechen
- aber idR mehr Bits enthalten
- in Instruktionen dennoch direkt genutzt werden können.

Sie müssen in reale Adressen umgesetzt werden, um tatsächliche Speicherinhalte anzusprechen.

## Analogie: Vollasoziativer Cache

In gewissem Sinne bereits dort: Reale Adresse muss erst auf den richtigen Cache-Block umgesetzt werden.

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 15/66**

# Von Cache-Blocks zu virtuellen Speicherseiten

- Ähnlich wie bei Cache Blocks: Wir fassen aufeinander folgende virtuelle Adressen zusammen und behandeln sie gleich bei der Umsetzung

## Definition 16.3 (Seite im virtuelle Speicher)

- Eine Seite (page) im virtuellen Speicher sind Speicherinhalte an aufeinander folgenden virtuelle Adressen.
- Seiten haben eine feste Größe (typisch etwas 4KiB bis 2MiB); in der Regel eine Zweierpotenz
- Anfangsadresse einer Seite ist ein Vielfaches dieser Größe (Alignment)

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 16/66**

- Gegenstück zu Seiten: Kacheln mit **realen** Adressen

## Definition 16.4 (Kachel im realen Arbeitsspeicher)

- Eine Kachel (frame) im **realen** Speicher sind Speicherinhalte an aufeinander folgenden **realen** Adressen
- Kacheln haben eine feste Größe; in der Regel identisch mit Seitengröße
- Anfangsadresse einer Kachel ist ein Vielfaches dieser Größe (Alignment)

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 17/66**

# Verallgemeinerung: Kachel in irgendeinen realen Speicher

- Mit Def 16.4 haben wir noch keine **zusätzlichen** Adressen gewonnen
- Daher Verallgemeinerung: Kachel darf in **irgendeinem realen** Speicher existieren
  - Arbeitsspeicher, Hauptspeicher

## Definition 16.5 (Kachel in realem Speicher)

- Wie Definition 16.4, aber Kachel kann in irgendeinem realen Speicher existieren
- Adressen können je nach Speicher unterschiedliche Formate haben

Hauptspeicher

Virtueller Speicher

**Virtuelle Adressen**

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 18/66**

# Inhaltsverzeichnis

---

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

**2.2 Adressumsetzung**

2.3 Seitentabelle

2.4 MMU und TLB

2.5 Seitenfehler

2.6 Verdrängung

2.7 Zusammenfassung

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

**Adressumsetzung**

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 19/66**



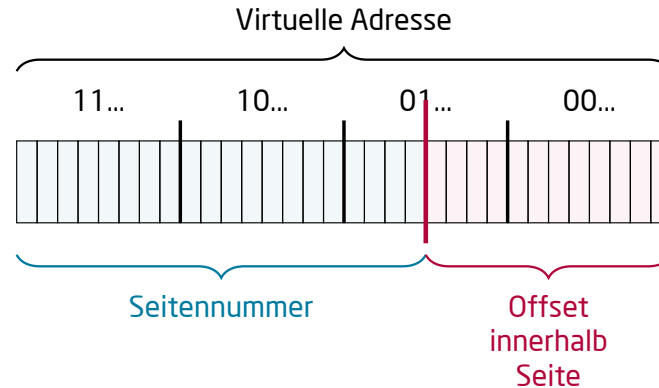
# Struktur virtueller Adressen

Wir machten Annahme:

- Seiten haben feste Größe
- Größe ist Zweierpotenz

Für virtuelle Adressen: Aufteilung in Bits

- Innerhalb einer Seite: **Offset**
  - Anzahl:  $\log_2$  der Seitengröße
- Welche Seite: **Seitennummer**



**Abbildung 16.4:** Aufteilung einer virtuellen Adresse in Offset und Seitennummer; Beispiel 32 Bits insgesamt; 12 Bits für Offset für Seitengröße 4 KiB; restliche 20 Bits nummerieren die Seiten durch

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller Speicher**

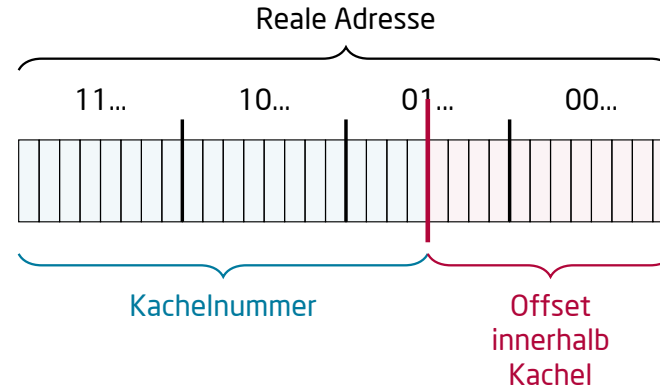
H. Karl, WS 22/23

**Folie 20/66**

# Reale Adressen

Gleiche Struktur:

- Offset, gleiche Länge wie bei virtueller Adresse
  - Das geht nicht anders!
- Kachelnummer
  - Kann kürzer oder länger als Seitennummer sein; siehe unten



**Abbildung 16.5:** Aufteilung einer realen Adresse in Offset und Kachelnummer; hier gleiche Aufteilung wie bei virtueller Adresse

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 21/66**

# Unterschiedliche Adressgrößen?

---

- Bei gleicher Größe virtuell  $\leftrightarrow$  real  
haben wir noch nicht viel gewonnen

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 22/66**

# Unterschiedliche Adressgrößen?

- Bei gleicher Größe virtuell  $\leftrightarrow$  real haben wir noch nicht viel gewonnen
- Reale Adresse (im Hauptspeicher) länger als virtuelle Adresse?
  - Das geht, aber dann kann ein Programm nicht alle möglichen Adressen nutzen
  - Wenig nützlich

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

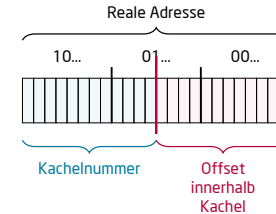
**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 22/66**

# Unterschiedliche Adressgrößen?

- Bei gleicher Größe virtuell  $\leftrightarrow$  real haben wir noch nicht viel gewonnen
- Reale Adresse (im Hauptspeicher) länger als virtuelle Adresse?
  - ☐ Das geht, aber dann kann ein Programm nicht alle möglichen Adressen nutzen
  - ☐ Wenig nützlich
- Reale Adresse kürzer als virtuelle Adresse?
  - ☐ Sinnvoll!



**Abbildung 16.6:** Reale Adresse für Hauptspeicher kürzer als virtuelle Adresse

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

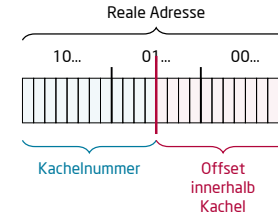
**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

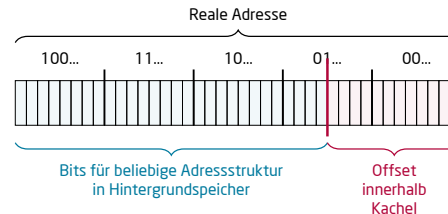
**Folie 22/66**

# Unterschiedliche Adressgrößen?

- Bei gleicher Größe virtuell ↔ real haben wir noch nicht viel gewonnen
- Reale Adresse (im Hauptspeicher) länger als virtuelle Adresse?
  - ☐ Das geht, aber dann kann ein Programm nicht alle möglichen Adressen nutzen
  - ☐ Wenig nützlich
- Reale Adresse kürzer als virtuelle Adresse?
  - ☐ Sinnvoll!
- Aber: Um alle virtuelle Adressen abzubilden brauchen wir Adresse in Hintergrundpeicher



**Abbildung 16.6:** Reale Adresse für Hauptspeicher kürzer als virtuelle Adresse



**Abbildung 16.7:** Reale Adresse für Hintergrundpeicher länger als virtuelle Adresse

Hintergrundpeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

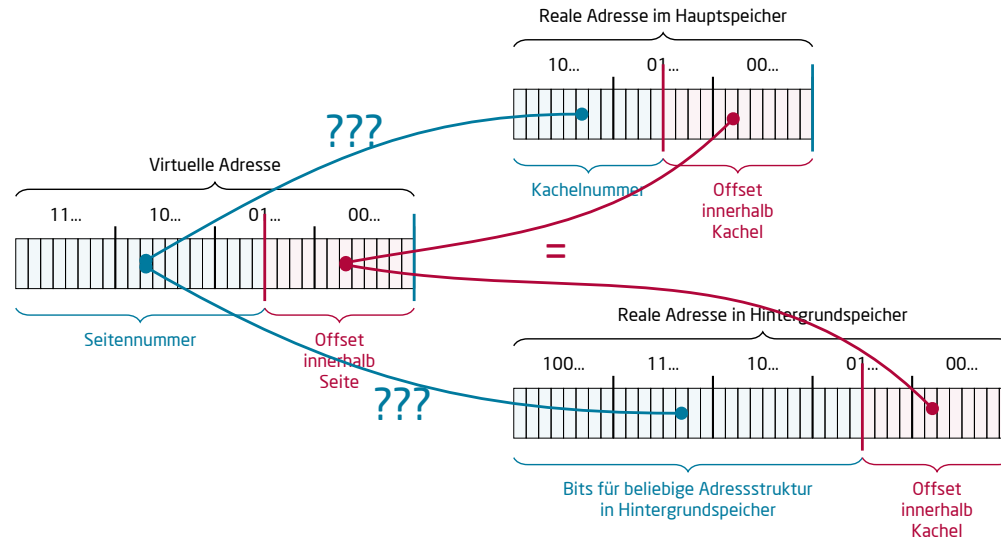
**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 22/66**

# Bezug virtuelle und unterschiedliche reale Adressen

- Offset identisch zwischen allen Versionen einer Adresse!
- Seitennummer muss auf die richtige Kachelnummer abgebildet werden



**Abbildung 16.8:** Zusammenhang zwischen Seitenadressen und Kacheladressen in Haupt- bzw. Hintergrundspeicher

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 23/66**

# Inhaltsverzeichnis

---

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

2.2 Adressumsetzung

**2.3 Seitentabelle**

2.4 MMU und TLB

2.5 Seitenfehler

2.6 Verdrängung

2.7 Zusammenfassung

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 24/66**



# Bezug zwischen Seitennummern und Kachelnummern

- Annahme:
  - ☐ Mehr Seiten als Kacheln im Hauptspeicher
  - ☐ Weniger Seiten als Kacheln im Hintergrundspeicher
- Wie finden wir für Seitennummer die richtige Kachelnummer?

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 25/66**

# Entwurfsentscheidung: Kopie?

Kann es sein, dass eine Seite

- Nur im Hintergrundspeicher, aber nicht im Hauptspeicher existiert?
  - ☐ Ja - dazu erfinden wir das ja
- Nur in Hauptspeicher, nicht im Hintergrundspeicher existiert?
  - ☐ Ja - wenn wir eine Seite neu erzeugen wird das im Hauptspeicher passieren
- In beiden realen Speichern existiert?
  - ☐ Ja - aber nicht unbedingt gleich aktuell
- In keinem realen Speicher existiert?
  - ☐ Ja - wenn wir von dieser Seite keine Adressen benutzt haben

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 26/66**

# Entwurfsentscheidung: Kopie?

Kann es sein, dass eine Seite

- Nur im Hintergrundspeicher, aber nicht im Hauptspeicher existiert?
  - ☐ Ja - dazu erfinden wir das ja
- Nur in Hauptspeicher, nicht im Hintergrundspeicher existiert?
  - ☐ Ja - wenn wir eine Seite neu erzeugen wird das im Hauptspeicher passieren
- In beiden realen Speichern existiert?
  - ☐ Ja - aber nicht unbedingt gleich aktuell
- In keinem realen Speicher existiert?
  - ☐ Ja - wenn wir von dieser Seite keine Adressen benutzt haben

Kacheln in Haupt- und Hintergrundspeicher sind **keine** direkten Kopien voneinander!

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

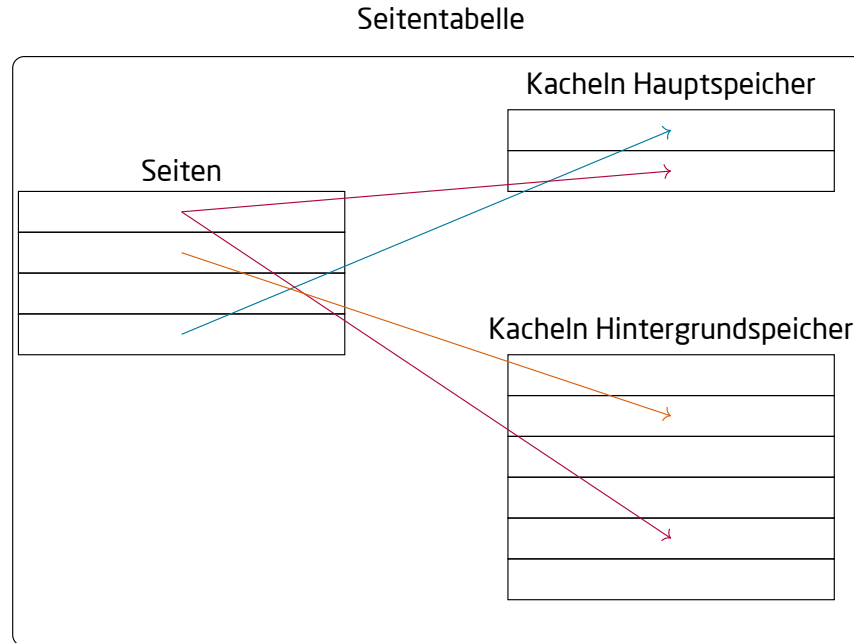
Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 26/66**

# Vier Kombinationen: Kachel in Haupt- und Hintergrundspeicher



**Abbildung 16.9:** Seiten können gar nicht existieren, nur auf einer Kachel im Hauptspeicher (blauer Pfeil), nur auf einer Kachel im Hintergrundspeicher (oranger Pfeil), auf Kachel in Haupt- und Hintergrundspeicher (roter Pfeil, möglicherweise mit unterschiedlicher Aktualität)

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 27/66**

# Analogie: Vollassoziativer Cache

- Analog zu Bezug realer Adresse und Cache-Block!
- Für volle Flexibilität: Vollassoziativer Ansatz, jede Seite kann auf jede Kachel abgebildet werden

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 28/66**

# Abbildung Seite auf Kachel: Seitentabelle

Für volle Flexibilität: Beliebige Seitennummern auf beliebige Kachelnummern abbilden

- **Seitentabelle** (page table)
- Offsets müssen **nicht** repräsentiert werden; die bleiben stets gleich!

**Tabelle 16.1:** Struktur einer Seitentabelle (page table) ohne Verwaltungsinformation

Seitennummer	Kachelnummer in Hauptspeicher	Kachelnummer in Hintergrundspeicher
0	17	42
1	19	--
2	--	--
3	--	--
4	28	43
5	--	46
...	...	...

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 29/66**

# Verwaltungsinformation in Seitentabelle

Was müssen wir über eine Seite (eine Zeile in der Seitentabelle) wissen?

## Unverzichtbar

- Ist die Seite im Hauptspeicher? (ist die entspr. Kachelnummer gültig?)
  - ☐ Sog. **present**-Bit
- Ist die Seite im Hintergrundspeicher? (ist die entspr. Kachelnummer gültig?)
- Falls beides: Version im Hauptspeicher aktueller als die im Hintergrundspeicher?
  - ☐ Durch Schreiboperation
  - ☐ Sog. **dirty**-Bit

## Optional

- Enthält die Seite Code oder Daten?
  - ☐ Darf der PC auf diese Seite zeigen?
- Wurde auf die Seite schon zugegriffen?
  - ☐ Sog. **reference**-Bit
- Schutzrechte für die Seite?
  - ☐ Z.B. Schreibschutz
- Caching für diese Seite erlaubt?
  - ☐ Wichtig für Daten, die zwischen Prozessen geteilt sind

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 30/66**

# Praktische Darstellung der Seitentabelle

## Konzept:

- Bisher: Konzeptionelle Sicht auf Seitentabelle
- Praktische Realisierung muss kompakt, schnell zugreifbar sein

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 31/66**



# Praktische Darstellung der Seitentabelle

## Konzept:

- Bisher: Konzeptionelle Sicht auf Seitentabelle
- Praktische Realisierung muss kompakt, schnell zugreifbar sein

## Realisierung:

- Information über Hintergrundspeicher wird relativ selten gebraucht
  - Meist in separate Tabelle ausgelagert
- Nicht alle Zeilen sind überhaupt relevant; viele Seiten von Programmen nicht benutzt
  - Kompakte Darstellung durch hierarchische oder invertierte Seitentabellen

Vgl. VL Betriebssysteme!

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 31/66**

# Inhaltsverzeichnis

---

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

2.2 Adressumsetzung

2.3 Seitentabelle

**2.4 MMU und TLB**

2.5 Seitenfehler

2.6 Verdrängung

2.7 Zusammenfassung

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

**MMU und TLB**

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 32/66**

# Adressumsetzung in Software?

- Programm soll **nur** virtuelle Adressen benutzen
- Umrechnung passiert bei **jedem** Speicherzugriff
- Muss Programm das selbst machen?
  - Also statt einfach `ldw (t0)`, 4096 müssten wir hinschreiben:
    - Extrahiere bits der Seitennummer
    - Schlage in Tabelle nach (das sind auch wieder Speicherzugriffe!!); entnimm Kachelnummer
    - Baue neue reale Adresse zusammen
    - **Dann** erst `ldw ...`

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 33/66**

# Adressumsetzung in Software?

- Programm soll **nur** virtuelle Adressen benutzen
- Umrechnung passiert bei **jedem** Speicherzugriff
- Muss Programm das selbst machen?
  - Also statt einfach `ldw (t0)`, 4096 müssten wir hinschreiben:
    - Extrahiere bits der Seitennummer
    - Schlage in Tabelle nach (das sind auch wieder Speicherzugriffe!!); entnimm Kachelnummer
    - Baue neue reale Adresse zusammen
    - **Dann** erst `ldw ...`
- Ganz sicher nicht praktikabel
  - Wir brauchen Hardware-Support dafür!

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

**MMU und TLB**

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 33/66**

# Memory Management Unit (MMU)

- Hardware-Support für Adressumsetzung: Memory Management Unit
- Verfahren wie auf vorheriger Folie
- Architektur: MMU zwischen Cache und Hauptspeicher

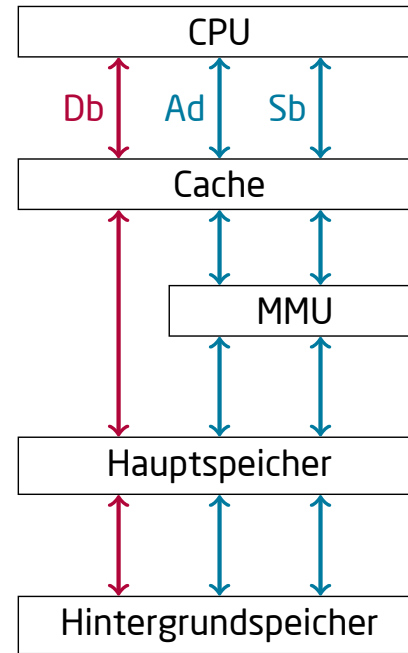


Abbildung 16.10: MMU zwischen Cache und Hauptspeicher

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

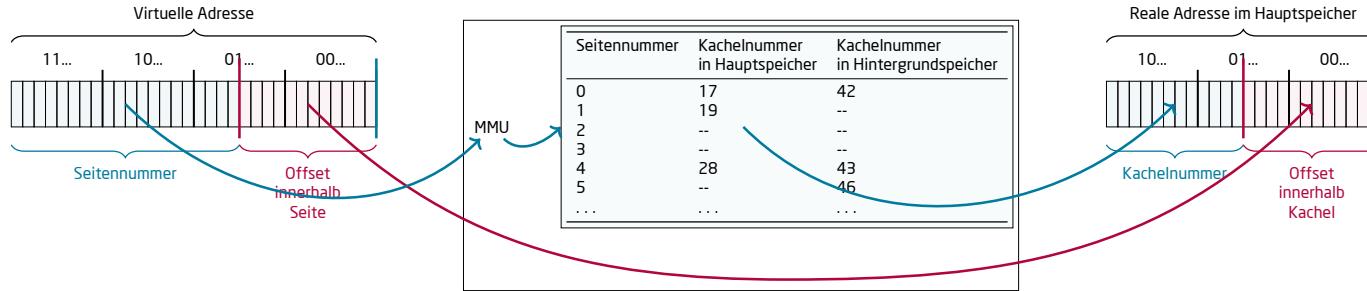
Material

**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 34/66**

# Adressumsetzung durch MMU



**Abbildung 16.11:** Umsetzung einer virtuellen Adresse in reale Adresse (im Beispiel: im Hauptspeicher) durch Aufteilen in Seitennummer und Offset, Nachschlagen der Seitennummer in Seitentabelle und Finden der Kachelnummer, und Hinzufügen der des Offsets zur Kachelnummer

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller Speicher**

H. Karl, WS 22/23

**Folie 35/66**

# Speicherzugriffe bei Adressumsetzung?

- Zusätzlich zum eigentlichen Speicherzugriff: Zugriff auf Seitentabelle
- Organisation zur Minimierung der Anzahl Zugriff
  - Adresse der Zeile direkt aus Seitennummer berechenbar
  - Dann mindestens ein Zugriff
  - Aber Seitentabelle wird dann sehr groß; platzsparender Organisation braucht mehr Zugriff
- Und: Speicherzugriffe sind entscheidender Geschwindigkeitsengpass
  - Deswegen Caches, mehrstufig, . . .
- Seitentabelle (nur) im Hauptspeicher abzulegen ist **nicht akzeptabel**

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 36/66**

- Wir brauchen einen Cache für die Seitentabelle
  - Den normalen Cache können wir dafür nicht nehmen (vgl. Abb. 16.10)
- MMU bekommt einen eigenen Cache: den **Translation Look-Aside Buffer (TLB)**
  - Integration mit Prozesswechsel dabei kritisch; vgl. VL Betriebssysteme

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 37/66**



# MMU: Weitere Aufgaben

---

- Überwachung von Zugriffsrechten
  - Z.B. Instruktion von Daten-Seiten laden
- Schreibschutz von Seiten sicherstellen
- TLB zum richtigen Zeitpunkt leeren
- ...

Führen jeweils zum Auslösen weiterer Interrupts; Betriebssystem entscheidet

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 38/66**

# Inhaltsverzeichnis

---

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

2.2 Adressumsetzung

2.3 Seitentabelle

2.4 MMU und TLB

**2.5 Seitenfehler**

2.6 Verdrängung

2.7 Zusammenfassung

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

**Seitenfehler**

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 39/66**

# Zugriff auf Seite nicht im Hauptspeicher?

- Was passiert, wenn auf eine Seite zugegriffen werden soll, die **nicht** im Hauptspeicher vorhanden ist?
  - MMU kann das natürlich feststellen
  - Ein **Seitenfehler** tritt auf
  - Analog: Cache Miss

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

**Seitenfehler**

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 40/66**

# Wo Seitenfehler behandeln?

- Option 1: Wir erweitern MMU, um Seitenfehler zu behandeln
  - Denkbar, aber kompliziert
- Option 2: Wir behandeln Seitenfehler in Software
  - MMU löst Interrupt aus
  - Interrupt Service Routine (ISR) kümmert sich um Seitenfehler

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 41/66**

# Wo Seitenfehler behandeln: Abwägung?

- Seitenfehler führt (meist) dazu, dass die zugehörige Kachel von Hintergrundspeicher in irgendeine Kachel im Hauptspeicher kopiert werden muss
- Hintergrundspeicher ist ohnehin **laaaaaaaangsam**
- Option 1 lohnt nicht!
  - Mögliche kleine Geschwindigkeitsvorteil geht ohnehin durch langsamen Hintergrundspeicher verloren
  - Dafür schränkt man Flexibilität erheblich ein
  - Also: Software/Betriebssystem!
- Hier wesentlich andere Abwägung als bei Cache Miss
  - Abstand Cache vs. Hauptspeicher **viel** kleiner als Hauptspeicher vs. Hintergrundspeicher

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 42/66**

# Wie Seitenfehler behandeln?

Seitenfehler wird durch MMU ausgelöst, ISR des Betriebssystems läuft an

- Eingabe: Welche Seitennummer; Zugriff auf Seitentabelle
- Fall 1: Seite existiert im Hintergrundspeicher; siehe unten

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 43/66**

# Wie Seitenfehler behandeln?

Seitenfehler wird durch MMU ausgelöst, ISR des Betriebssystems läuft an

- Eingabe: Welche Seitennummer; Zugriff auf Seitentabelle
- Fall 1: Seite existiert im Hintergrundspeicher; siehe unten
- Fall 2: Seite existiert nicht
  - Also erster Zugriff auf eine virtuelle Adresse dieser Seite
  - Reaktion abhängig von Betriebssystem, Laufzeitumgebung, ggf. sogar Programmiersprache → VL Betriebssysteme

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 43/66**

# Seitenfehler: Seite existiert im Hintergrundspeicher

- Fall 1.1: Es gibt noch freie Kacheln im Hauptspeicher
  - Seite **einlagern**:
    - Kopie der Hintergrundspeicher-Kachel in Hauptspeicher erstellen
    - In Seitentabelle eintragen
    - Speicherzugriff erneut durchführen
    - Währenddessen: Programm angehalten!!

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 44/66**



# Seitenfehler: Seite existiert im Hintergrundspeicher

- Fall 1.1: Es gibt noch freie Kacheln im Hauptspeicher
  - Seite **einlagern**:
    - Kopie der Hintergrundspeicher-Kachel in Hauptspeicher erstellen
    - In Seitentabelle eintragen
    - Speicherzugriff erneut durchführen
    - Währenddessen: Programm angehalten!!
- Fall 1.2: Keine freie Kachel
  - Für freie Kachel sorgen: **Verdrängen** einer anderen Seite von ihrer Hauptspeicher-Kachel
  - Dann Speicherzugriff erneut durchführen (wir sind dann in Fall 1.1!)

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

**Seitenfehler**

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 44/66**

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

2.2 Adressumsetzung

2.3 Seitentabelle

2.4 MMU und TLB

2.5 Seitenfehler

**2.6 Verdrängung**

2.7 Zusammenfassung

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

**Verdrängung**

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 45/66**

# Verdrängen einer Seite

- Situation:
  - ☐ Alle Kacheln des Hauptspeichers sind mit Seiten belegt
  - ☐ Wir brauchen eine freie Kachel, um eine andere Seite aus Hintergrundspeicher einzulagern
- Welche Seite wählen wir zur **Verdrängung** aus?

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 46/66**

# Verdrängung: Optionen

## ■ Modifikationsbit berücksichtigen?

- ☐ Nicht modifizierte Seiten (die Kopie in Hintergrundspeicher haben) kann man einfach überschreiben
- ☐ Modifizierte Seiten müssen erst in Hintergrundspeicher geschrieben werden (Write back!)
- ☐ Nachteil: vielleicht werden die nicht modifizierten Seiten besonders oft gebraucht, weil dort Code steht?

## ■ Zugriffe berücksichtigen?

- ☐ Ähnlich zu Cache: Least Frequently Used (LFU), Least Recently Used (LRU) (und Varianten) sind denkbare Strategien
- ☐ Meist: Approximation von LRU
- ☐ Details: VL Betriebssysteme

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 47/66**

1. Hintergrundspeicher

**2. Virtueller Speicher**

2.1 Virtuelle Adressen

2.2 Adressumsetzung

2.3 Seitentabelle

2.4 MMU und TLB

2.5 Seitenfehler

2.6 Verdrängung

**2.7 Zusammenfassung**

3. Programmausführung

4. Zusammenfassung

5. Material

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

**Zusammenfassung**

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 48/66**

# Same same, but different

- Ideen extrem ähnlich:
  - Cache für Hauptspeicher
  - Virtualisierung für Hauptspeicher durch Hintergrundspeicher
- Aus Perspektive Hauptspeicher: Erweiterung Speicherhierarchie nach oben oder unten
- Natürlich Unterschied im Detail: Mechanismus und Strategie

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 49/66**

# Größe von Seiten/Kacheln?

---

- Warum sind Seiten und Kacheln so viel größer als Blöcke im Cache?

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 50/66**

# Größe von Seiten/Kacheln?

- Warum sind Seiten und Kacheln so viel größer als Blöcke im Cache?

Mehrere Gründe:

- Die Miss Penalty ist um ein Vielfaches höher
- Der Aufwand für Verdrängung und Nachschub ebenso
- Insbes. Zeit, das erste Byte zu beschaffen
  - Latenz ist problematisch, Datenrate weniger

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 50/66**



# Größe von Seiten/Kacheln?

- Warum sind Seiten und Kacheln so viel größer als Blöcke im Cache?

Mehrere Gründe:

- Die Miss Penalty ist um ein Vielfaches höher
- Der Aufwand für Verdrängung und Nachschub ebenso
- Insbes. Zeit, das erste Byte zu beschaffen
  - Latenz ist problematisch, Datenrate weniger

Also: Wir amortisieren den Aufwand durch größere Einheiten

Hintergrundspeicher

Virtueller Speicher

Virtuelle Adressen

Adressumsetzung

Seitentabelle

MMU und TLB

Seitenfehler

Verdrängung

Zusammenfassung

Programmausführung

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 50/66**

# Inhaltsverzeichnis

---

- 1. Hintergrundspeicher
- 2. Virtueller Speicher
- 3. Programmausführung
  - 3.1 Mehrere Prozesse
  - 3.2 Speicherschutz
- 4. Zusammenfassung
- 5. Material

Hintergrundspeicher

Virtueller Speicher

**Programmausführung**

Mehrere Prozesse

Speicherschutz

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 51/66**

# Von Programm zu Prozess

---

- Bisher: nur ein einziges laufendes Programm innerhalb eines Rechners
  - Betriebssystem kam nur durch Interrupts, expliziter Aufruf zum Ablauf
- Wie führen wir mehrere Programme (scheinbar?) gleichzeitig aus?
  - Unterscheidung: Aus einem **statischen Programm** wird ein **dynamisch ablaufender Prozess**

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Mehrere Prozesse

---

- Problem: Welche Adressen verwenden die einzelnen Prozesse?
  - Unterschiedliche virtuelle Adressen? Wer legt das fest?
  - Wenn das gleiche Programme mehrmals gestartet wird, warum sollten das unterschiedliche Adressen sein???

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Mehrere Prozesse

---

- Problem: Welche Adressen verwenden die einzelnen Prozesse?
  - Unterschiedliche virtuelle Adressen? Wer legt das fest?
  - Wenn das gleiche Programme mehrmals gestartet wird, warum sollten das unterschiedliche Adressen sein???
- Forderung: Wir müssen **gleiche virtuelle Adressen** von **unterschiedlichen Prozessen** auf **unterschiedliche reale Adressen** umsetzen

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
**Mehrere Prozesse**  
Speicherschutz  
Zusammenfassung  
Material

# Eine Seitentabelle pro Prozess

---

Einfachste Idee: Jeder Prozess hat eine eigene Seitentabelle!

- Kann dann beliebige virtuelle Adressen benutzen
  - z.B. alle Prozesse haben Programmbeginn bei Adresse 0x100, . . .
- Sprechweise: Jeder Prozess besitzt seinen eigenen **virtuellen Adressraum**

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

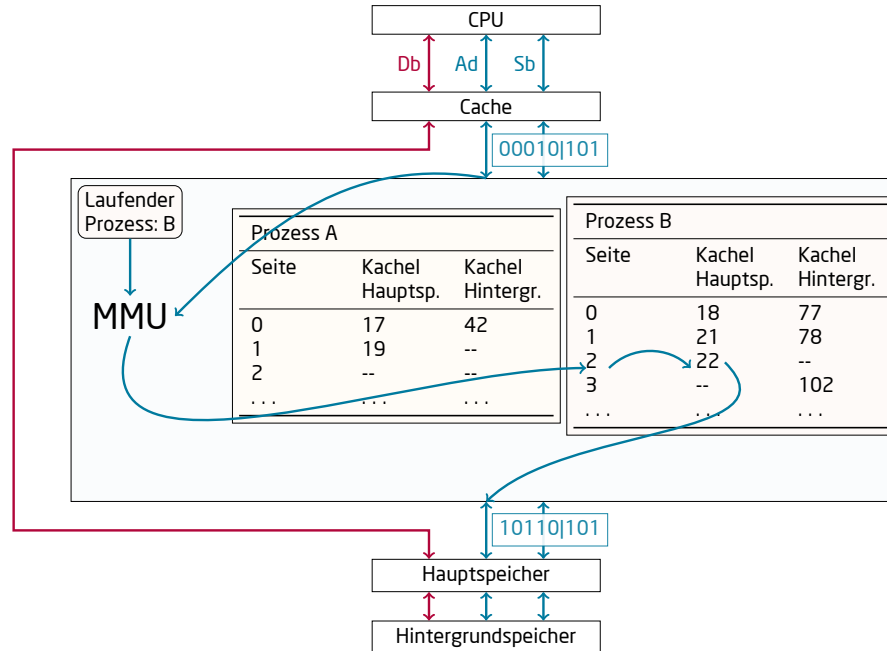
# Eine Seitentabelle pro Prozess: Anforderung

---

- Beim Wechsel von Prozess zu Prozess muss die **MMU** informiert werden und die benutzte Seitentabelle austauschen
  - TLB leeren!
- Bei Suche nach freien Kacheln müssen **alle** Seitentabellen berücksichtigt werden

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Adressauflösung in MMU mit mehreren Seitentabellen



**Abbildung 16.12:** Vergrößerung von Abb. 16.10: Prozess B greift auf virtuelle Adresse 00010|101 zu (3 Bits für Offset); MMU schaut die Seitennummer 00010 in Seitentabelle von Prozess B nach; findet Kachel 22, erzeugt reale Adresse 10110|101 (Kachelnummer und Offset angehängt); gibt diese Adresse auf den Adressbus zum Hauptspeicher; Cache und Hauptspeicher sehen davon nichts (transparent).

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material



# Inhaltsverzeichnis

---

- 1. Hintergrundspeicher
- 2. Virtueller Speicher
- 3. Programmausführung**
  - 3.1 Mehrere Prozesse
  - 3.2 Speicherschutz**
- 4. Zusammenfassung
- 5. Material

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Adressen in Programmen

---

Eine Konsequenz aus Abb. 16.10, Abb. 16.12:

- Laufender Prozess kann nur mit virtuellen Adressen arbeiten
- **Jeder** Zugriff auf Adressen wird durch MMU abgefangen und mit Seitentabelle umgerechnet

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Mehrere Prozesse

Speicherschutz

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 58/66**

# Zugriff auf Speicher anderer Prozesse

---

- Damit (an sich) kein Zugriff auf Speicher anderer Prozesse möglich (lesend oder schreibend)
  - Es sei denn, die Seitentabellen sind nicht korrekt gefüllt
  - Z.B. darf kein Eintrag in einer Seitentabelle auf eine Kachel zeigen, die Seitentabellen an sich speichert!

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Zugriff auf Speicher anderer Prozesse

- Damit (an sich) kein Zugriff auf Speicher anderer Prozesse möglich (lesend oder schreibend)
  - ☐ Es sei denn, die Seitentabellen sind nicht korrekt gefüllt
  - ☐ Z.B. darf kein Eintrag in einer Seitentabelle auf eine Kachel zeigen, die Seitentabellen an sich speichert!
- An sich. . .
  - ☐ Natürlich gibt es Angriffe gegen diesen Mechanismus (z.B. Meltdown)
  - ☐ Und Bugs in Betriebssystemen (z.B. Windows-Versionen, bei denen Prozesse Seitentabelle manipulieren konnten)
  - ☐ Vgl. VL Betriebssystem, Vorlesungen zu Sicherheit

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Mehrere Prozesse  
Speicherschutz  
Zusammenfassung  
Material

# Wer füllt Seitentabellen?

- Prozess selbst darf / kann seine Seitentabelle nicht manipulieren
- Muss also von "nicht-normalem" Prozess geschehen: Betriebssystem
  - Genauer: Code der in privilegiertem CPU-Zustand ausgeführt wird
- Konsequenz: Privilegierter Code kann an MMU vorbei mit realen Adressen arbeiten
  - Z.B. Seitentabellen normaler Prozesse manipulieren aus page fault handler heraus
  - Z.B. Geräte-Register lesen/schreiben (vgl. Folie 7.23 ff.)

Hintergrundspeicher

Virtueller Speicher

Programmausführung

Mehrere Prozesse

Speicherschutz

Zusammenfassung

Material

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 60/66**

# Inhaltsverzeichnis

---

- 1. Hintergrundspeicher
- 2. Virtueller Speicher
- 3. Programmausführung
- 4. Zusammenfassung**
- 5. Material

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
**Zusammenfassung**  
Material

- Virtualisierung: Mangelverbergung ohne Änderung der Nutzungsweise
- Virtualisierung von Hauptspeicher:
  - Mangel an Hauptspeicher durch Hintergrundspeicher verborgen
  - Ermöglicht einfaches Konzept zur nebenläufigen Ausführung von Programmen in einem System
- Viele Ideen ähnlich zu Cache für Hauptspeicher
  - Aber signifikante Unterschiede in absoluten Zugriffszeiten / Verhältnissen zwischen Zeiten, Größenordnungen
  - Erfordert andere Entwurfsentscheidungen, Strategien

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
**Zusammenfassung**  
Material

# Inhaltsverzeichnis

---

- 1. Hintergrundspeicher
- 2. Virtueller Speicher
- 3. Programmausführung
- 4. Zusammenfassung
- 5. Material

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
Material



Hintergrundspeicher

Virtueller Speicher

Programmausführung

Zusammenfassung

**Material**

**GDS 16: Virtueller  
Speicher**

H. Karl, WS 22/23

**Folie 64/66**

**ISR** Ein Unterprogramm, das für die Behandlung einer Unterbrechung geeignet und gedacht ist. 52

**LFU** Am seltensten genutzt -- Strategie, bei der bevorzugt selten genutzte Einträge aus dem Cache verdrängt werden. 60

**LRU** Am wenigsten kürzlich genutzt -- Strategie, bei der bevorzugt lange nicht benutzte Einträge aus einem Cache verdrängt werden. 60

**MMU** Memory Management Unit 45

**TLB** Translation Look-Aside Buffer 48

**Kachel** In einem System mit virtuellem Speicher ist eine Kachel (engl. frame) der Inhalt einer Menge aufeinander folgender *realer* Adressen. Kachel haben idR. die gleiche Größe wie die Seiten des Systems. Die Anfangsadressen sind Vielfache dieser Größe (Alignment). Kacheln gibt es in unterschiedlichen Arten realen Speichers (Hauptspeicher, Hintergrundspeicher wie Festplatte). 22

**Memory Management Unit** Memory Management Unit: Hardware-Komponente, die virtuelle Adressen in reale Adressen umrechnet. 45

**Seite** In einem System mit virtuellem Speicher ist eine Seite (engl. page) der Inhalt einer Menge aufeinander folgender *virtueller* Adressen. Seiten haben eine feste Größe (typische Werte liegen zwischen 4 KiB und 2MiB je nach System; es sind praktisch immer Zweipotenzen). Die Anfangsadressen sind Vielfache dieser Seitengröße (Alignment). 21

Hintergrundspeicher  
Virtueller Speicher  
Programmausführung  
Zusammenfassung  
**Material**