

# 11. Tutorium

## Paging

Rechnerorganisation, Tutorium #13

Patrick Röper | 21. Januar 2020

FAKULTÄT FÜR INFORMATIK



## 1 Paging

## 2 Aufgaben

## ■ Segmentierung (variable Größe)

- logische Abbildung einer Programmstruktur
- geringer Datentransfer
- umfangreicher Datentransfer beim Ein-/Auslagern
- externe Fragmentierung

## ■ Seitenwechsel-Verfahren (feste Größe)

- geringerer Verwaltungsaufwand
- bessere Hauptspeicherauslastung
- häufiger Datentransfer
- interne Fragmentierung

- Hier muss eine ausreichend große Lücke im Hauptspeicher gefunden werden (bei Seitenverfahren sind hingegen alle Seiten gleich groß)
- Drei Strategien (Betriebssystem):
  - **first-fit:** erste passende Lücke wird genommen
  - **best-fit:** kleinste passende Lücke wird genommen
  - **worst-fit:** größte passende Lücke wird genommen

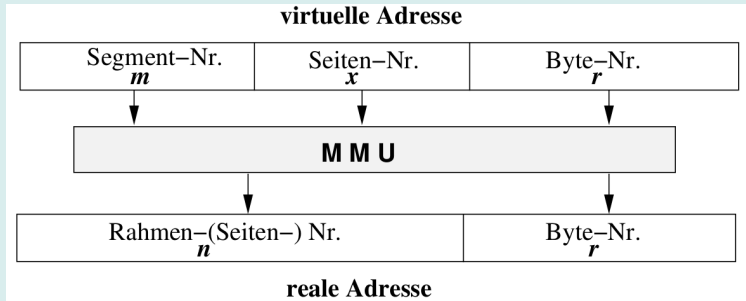
- 5 gängige Strategien zum Ersetzen einer Seite:
  - **FIFO (first-in-first-out):** die sich am längsten im Hauptspeicher befindende Seite wird ersetzt
  - **LIFO (last-in-first-out):** die zuletzt eingelagerte Seite wird ersetzt
  - **LRU (least recently used):** die Seite, auf die am längsten nicht zugegriffen wurde, wird ersetzt
  - **LFU (least frequently used):** die seit ihrer Einlagerung am seltensten benutzte Seite wird ersetzt
  - **LRD (least reference density):** Mischung aus LRU und LFU; die Seite mit der geringsten Zugriffsichte (Anzahl Zugriffe / Einlagerungszeitraum) wird ersetzt
- Daneben werden bevorzugt solche Seiten ersetzt, die nicht verändert wurden → kein Rückschreiben der geänderten Seite erforderlich



# Aufgabe

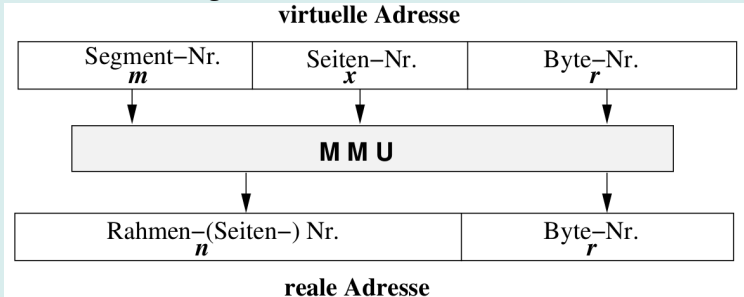
Die Speicherverwaltung in einem Rechnersystem geschieht zweistufig über eine Segmenttabelle und eine Seitentabelle. Die Unterteilung der virtuellen und der physikalischen Adresse ist unten dargestellt.

**Hinweis:** Zur Lösung dieser Aufgabe verwenden Sie bitte gängige Werte für  $m$ ,  $x$  und  $r$  aus der Literatur.



# Aufgabe

Geben Sie die Größe des maximal verfügbaren virtuellen Adressraums in Byte an. In wieviele Segmente wird der virtuelle Adressraum unterteilt?



m=10 Bit; x=10 Bit; r=12 Bit  
=> Adresse = 32 Bit



## Lösung

Größe des maximal verfügbaren virtuellen Adressraums in Byte:

$$2^{32} \text{ Byte} = 4 \text{ GByte} \quad 0 \dots (2^{32} - 1)$$

Anzahl der Segmente im virtuellen Adressraum:

$$2^{10} = 1024 \text{ Segmente} \quad 0 \dots 1023$$

## Aufgabe

Wieviel Seiten können in einem Segment im virtuellen Adressraum gespeichert werden? Geben Sie die Größe einer Seite in Byte an.

## Aufgabe

Wieviel Seiten können in einem Segment im virtuellen Adressraum gespeichert werden? Geben Sie die Größe einer Seite in Byte an.

## Lösung

Anzahl der Seiten pro Segment:

$$2^{10} = 1024 \text{ Seiten} \quad 0 \dots 1023$$

Größe einer Seite in Byte:

$$2^{12} \text{ Byte} = 4 \text{ Kbyte}$$

## Aufgabe

Welche Informationen enthält ein Segment-Deskriptor?

## Aufgabe

Welche Informationen enthält ein Segment-Deskriptor?

## Lösung

Informationen im Segment-Deskriptor:

- Informationen darüber, ob das Segment gültig ist und wieviele Seiten es umfasst.
- Welche Seitentabellen im Hauptspeicher präsent sind.
- Zeiger auf eine Seitentabelle, welche die Deskriptoren aller zu diesem Segment gehörenden Seiten und die Rahmennummer der Seiten enthält.

## Aufgabe

Was sind die Vorteile bzw. Nachteile einer solchen zweistufigen Adressumsetzung gegenüber einer reinen Seitenverwaltung?

## Aufgabe

Was sind die Vorteile bzw. Nachteile einer solchen zweistufigen Adressumsetzung gegenüber einer reinen Seitenverwaltung?

## Lösung

Vorteile bzw. Nachteile einer zweistufigen Adressumsetzung:

**Vorteil:** Bei einer reinen Seitenverwaltung hat man eine einzige, sehr große Seitentabelle, die im Hauptspeicher entsprechend viel Platz beansprucht. Im Gegensatz dazu gibt es bei einer zweistufigen Adressumsetzung viele, jedoch kleinere Seitentabellen, von denen nur die aktuelle im Hauptspeicher (neben der Segmenttabelle) gehalten werden muss. Die anderen Seitentabellen können in einem Hintergrundspeicher stehen und müssen dann bei Bedarf geladen werden.

**Nachteil:** Zweistufige Adressbildung.

## Aufgabe

Gegeben sei eine Speicherverwaltungseinheit (MMU). Der virtuelle Speicher ist in 8 Seiten mit je 1 KByte unterteilt. Der physikalische Speicher hat eine Kapazität von 4 KByte. Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 1 angegeben.

Virtuelle Seitennummer	Physikalische Seitennummer
0	-
1	-
2	1
3	3
4	-
5	0
6	2
7	-



## Aufgabe

Seitengröße: **1 KByte**

virtuelle Seiten: **8**

physikalische Speicherkapazität: **4 KByte**

Virtuelle Seitennummer	Physikalische Seitennummer
0	-
1	-
2	1
3	3
4	-
5	0
6	2
7	-

Skizzieren Sie die Unterteilung der 32-Bit breiten virtuellen Adresse.

# Aufgabe 2.1

## Aufgabe

Seitengröße: **1 KByte**

virtuelle Seiten: **8**

physikalische Speicherkapazität: **4 KByte**

Skizzieren Sie die Unterteilung der 32-Bit breiten virtuellen Adresse.

## Lösung

Unterteilung der virtuellen Adresse:



## Aufgabe

Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen:

**2100, 4095, 5620, 6200, 1023**

Virtuelle Seitennummer	Physikalische Seitennummer
0	-
1	-
2	1
3	3
4	-
5	0
6	2
7	-

## Aufgabe

Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen:

**2100, 4095, 5620, 6200, 1023**

## Lösung

Physikalische Adressen:

Virtuelle Adresse	Physikalische Adresse
2100	1076 (0x434)
4095	4095 (0xFFFF)
5620	500 (0x1F4)
6200	2104 (0x838)
1023	<i>page fault</i>

## Aufgabe

Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen **Translation Lookaside Buffer** (TLB) erreicht?

## Aufgabe

Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen **Translation Lookaside Buffer** (TLB) erreicht?

## Lösung

Eine Beschleunigung der Adressumsetzung durch den TLB wird beim zweiten Zugriff auf eine Seite und solange die entsprechenden Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle aus dem TLB nicht verdrängt wurden.

## Aufgabe

Gegeben sei eine Speicherverwaltungseinheit (MMU) mit einer **Seitengröße von 1 KByte, 8 virtuellen Seiten und 4 physikalischen Seiten (Frames)**. Der aktuelle Ausschnitt der Seitentabelle ist in Tabelle 2 angegeben.

Virtuelle Seitennummer	Physikalische Seitennummer
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

## Aufgabe

Seitengröße: **1 KByte**

virtuelle Seiten: **8**

physikalische Seiten: **4**

Virtuelle Seitennummer	Physikalische Seitennummer
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

1. Skizzieren Sie die Unterteilung der **32 Bit** breiten virtuellen Adresse.

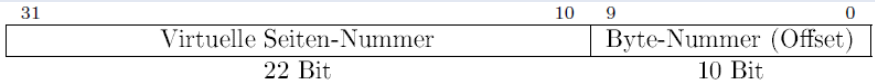


## Aufgabe

Skizzieren Sie die Unterteilung der 32 Bit breiten virtuellen Adresse.

## Lösung

Unterteilung der virtuellen Adresse:



## Aufgabe

Seitengröße: **1 KByte**

virtuelle Seiten: **8**

physikalische Seiten: **4**

Virtuelle Seitennummer	Physikalische Seitennummer
0	3
1	1
2	-
3	-
4	2
5	-
6	0
7	-

2. Ermitteln Sie die **physikalischen Adressen** zu den folgenden virtuellen Adressen:

1023; 1024; 4204; 6200

## Aufgabe

Ermitteln Sie die physikalischen Adressen zu den folgenden virtuellen Adressen:

1023; 1024; 4204; 6200

## Lösung

Physikalische Adressen:

Virtuelle		Physikalische	
Adresse	Seitennummer	Seitennummer	Adresse
1023	0	3	$3 * 1024 + 1023 = 4095$
1024	1	1	$1 * 1024 + 0 = 1024$
4204	4	2	$2 * 1024 + 108 = 2156$
6200	6	0	$0 * 1024 + 56 = 56$

## Aufgabe

Zur Beschleunigung der Adressberechnung soll ein Cache-Speicher als **Translation-Lookaside-Buffer (TLB)** eingesetzt werden, der die letzten **32 Einträge** aus dem Seitentabellen-Verzeichnis und der Seitentabelle speichert.

3. Unter welchen Bedingungen wird eine **Beschleunigung** der Adressumsetzung durch einen Translation Lookaside Buffer (TLB) erreicht?

## Aufgabe

Unter welchen Bedingungen wird eine Beschleunigung der Adressumsetzung durch einen Translation Lookaside Buffer (TLB) erreicht?

## Lösung

Eine Beschleunigung der Adressumsetzung durch den TLB wird beim **zweiten Zugriff** auf eine Seite und solange die entsprechenden Einträge aus dem Seitentabellen-Verzeichnis und der Seitentabelle aus dem TLB **nicht verdrängt** wurden.

## Aufgabe

4. Wie breit ist der **Tag** eines Cache-Eintrags?

Gehen Sie dabei von einer **n Bit breiten virtuellen Adresse**, einer **m Bit breiten physikalischen Adresse** und einer **Seitengröße von 4 KByte** aus.

## Aufgabe

Wie breit ist der Tag eines Cache-Eintrags?

Seitengröße: **4 KByte**

virtuelle Adresse: **n Bit**

physikalische Adresse: **m Bit**

## Lösung

Breite des Tags:

Seitengröße ist 4 KByte  $\Rightarrow 4096 \Rightarrow 2^{12} \Rightarrow$  Byte-Offset ist 12 Bit breit.

Der Tag ist dann  $(n - 12)$  Bits breit

# Was ihr jetzt kennen und können solltet...

- Grundlagen der virtuellen Speicherverwaltung
- Vorgehensweise bei der Segmentierung und dem Seitenwechselverfahren
- Übersetzung von virtuellen und physische Adressen



