



Betriebssysteme

Speicherverwaltung II: Seitenadressierung (Paging)



Überblick

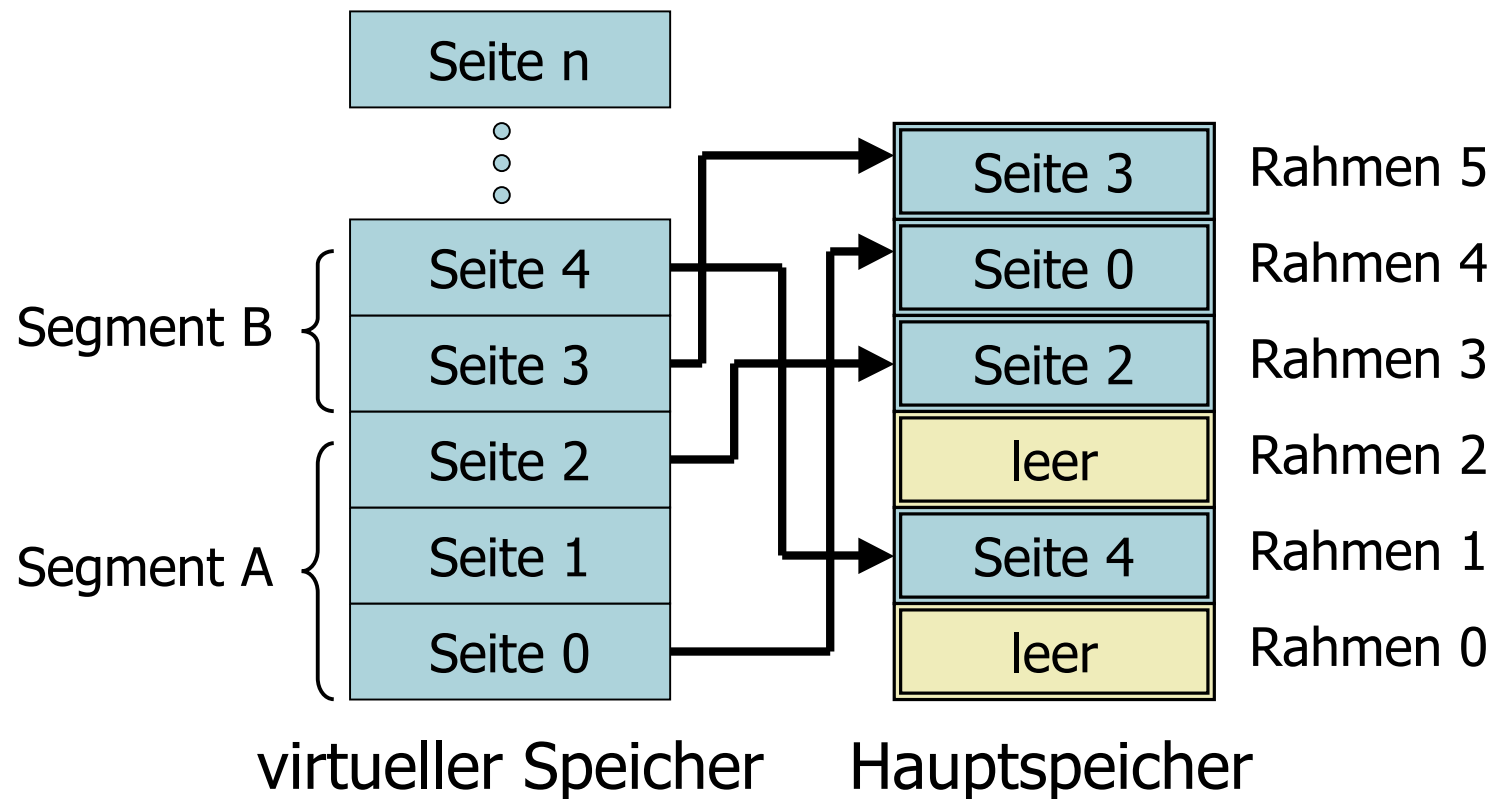
- Das Konzept der Seitenadressierung
- Seitentabellen
- Adresskonversionsverfahren
- Optimale Seitengröße
- Ersetzungsstrategien
- Vor- und Nachteile der Seitenadressierung



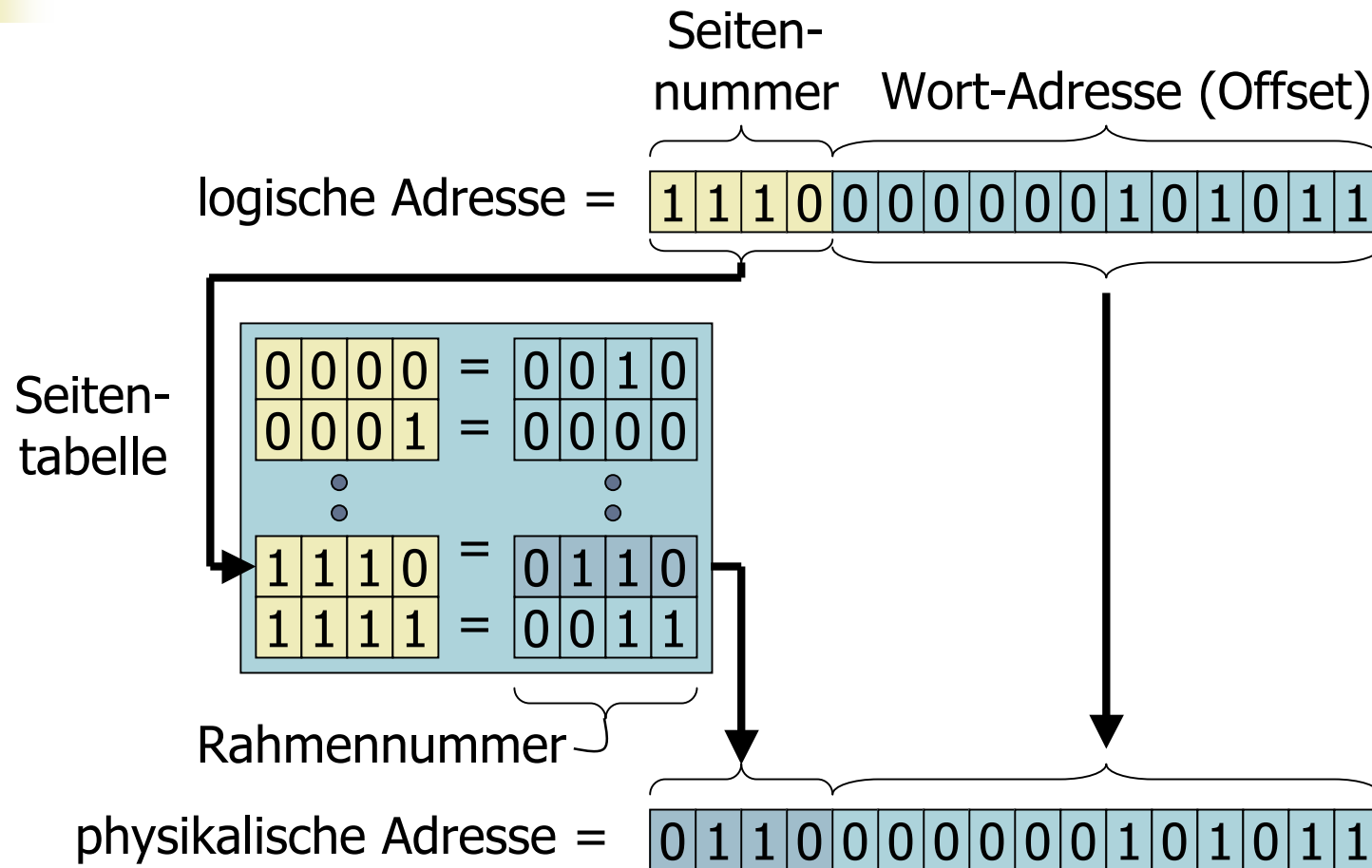
Seitenadressierung (paging)

- Unterteilung des logischen Adressraums in Seiten:
 - Abschnitte gleicher Größe (Seiten/pages)
 - Unterteilung des physikalischen Adressraums in korrespondierende Abschnitte (Rahmen/frames)
 - Aufteilung gemäß physikalischer Gesichtspunkte (leichte Adressierung, keine Fragmentierung)
 - typische Seiten-/Rahmen-Größe: 512 Byte – 8 KB

Zuordnung von Seiten zu Rahmen



Logische und reale Adressen bei Seitenadressierung





Seitentabellen

- Tabelle mit Informationen zu allen Seiten eines Prozesses:
 - Seitennummer
 - zugehörige Rahmennummer
 - Zugriffsattribute (für Zugriffsschutz)
 - Markierung geladen/nicht geladen
 - Markierung verändert/nicht verändert (dirty tag/modified bit)

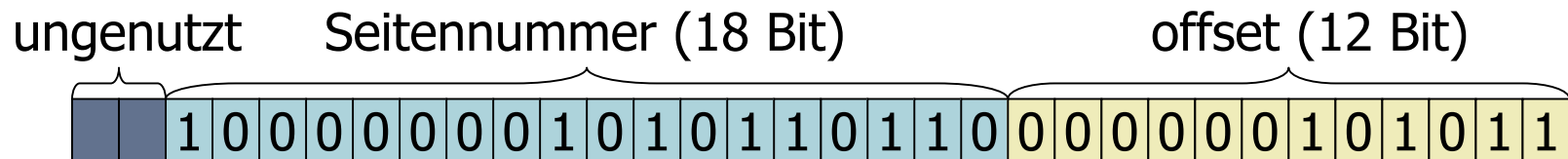


Adresskonversionsverfahren

- Problem: Direkte Abbildung der virtuellen Adresse auf die physikalische aufgrund der Größe der benötigten Seitentabelle nicht immer praktikabel
- Lösungsansätze:
 - Adressbegrenzung
 - Multi-level-Tabellen
 - Invertierte Seitentabellen
 - Assoziativer Tabellencache

Adressbegrenzung

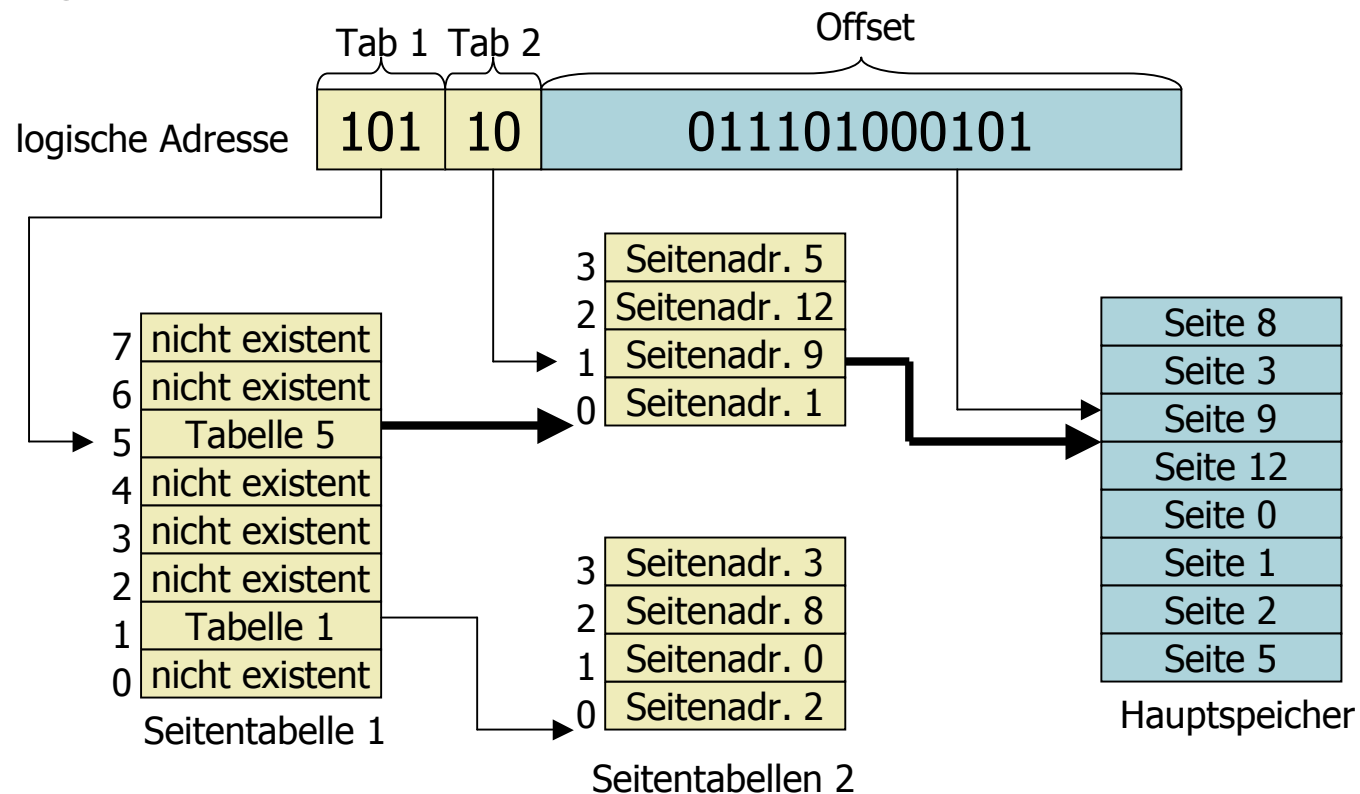
- Begrenzung des virtuellen Adressraums auf sinnvolle Maximalgröße



- Nachteile:
 - Begrenzung bedeutet eine Einschränkung
 - Seitentabellen für die meisten Prozesse dennoch zu groß

Multi-level-Tabellen

- Wird ein Speicherbereich überhaupt genutzt? (Tabelle 1)
- Falls ja, wie wird er benutzt? (Tabellen 2)



Invertierte Seitentabellen

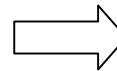
- Tabelle enthält nicht mehr Einträge für alle möglichen Seiten, sondern nur noch für die tatsächlich im Hauptspeicher befindlichen
- Invertierung: den Rahmennummern werden die Seitennummern zugeordnet

virtuell	reell
0	-
1	-
2	-
3	-
4	2
5	5
6	7
7	-

Seitentab. (Prozess 1)

virtuell	reell
0	-
1	-
2	0
3	-
4	-
5	1
6	-
7	4

Seitentab. (Prozess 2)



reell	virtuell	Proz. ID
0	2	2
1	5	2
2	4	1
3	-	-
4	7	2
5	5	1
6	-	-
7	6	1

Invertierte Seitentabelle

Assoziativer Tabellencache

- Pufferung der letzten Seiten/Rahmen-Zuordnungen in einem speziellen Cache (translation lookaside buffer, TLB)
- Der Cache ist ein Assoziativspeicher, erlaubt also inhaltsorientierten Zugriff

Assoziativ- speicher	Pr.ID		virtuell				reell				
	1	0	0	0	1	0	0	0	0		
	1	0	0	1	0	1	0	0	0	1	
	0	1	0	1	0	1	0	1	0	1	
	1	0	0	1	1	1	0	1	0	0	
	↑	↑	↑	↑	↑	↑	↓	↓	↓	↓	
Abfragewort	1	0	0	1	0	1	0	0	0	1	Antwort



Gemeinsam genutzter Speicher (shared memory)

- Dynamische, kontrollierte, gemeinsame Nutzung von Speicherbereichen
- Abbildung eines physikalischen Speicherbereichs in mehrere virtuelle Bereiche
- Notwendig: Freigabemechanismus, Löschschutz und Synchronisationsverfahren
- Beispiele:
 - Mehrfaches, gleichzeitiges Ausführen desselben Programms
 - Benutzung derselben Programm-Bibliotheken
 - Gleiche Datenbereiche (z. B. Parameter zur Fensterdarstellung)



Optimale Seitengröße

- Sei h die Hauptspeichergröße und s die Seitengröße
- Annahmen:
 - Die Prozessgröße ist gleichverteilt
 - Pro Prozess wird eine einstufige Seitentabelle mit $\lceil h / s \rceil$ Einträgen verwendet
 - Ein Eintrag in der Seitentabelle beansprucht ein Wort
- Folgerungen:
 - Der mittlere Verschnitt beträgt $s / 2$ Worte
 - Der mittlere Verlust V beträgt $(h / s + s / 2)$ Worte $\equiv h f_v$
 - Die optimale Seitengröße s_{opt} beträgt $\sqrt{2h}$ Worte
 - Der Verlustfaktor f_v beträgt in diesem Fall $2 / s_{\text{opt}}$



Seitenersetzungsstrategien

- **Problem:** Alle Rahmen im Hauptspeicher sind belegt, es muss aber eine neue Seite eingelagert werden → Auslagerung notwendig
- **Ziel:** Auswahl der auszulagernden Seiten, so dass (langfristig gesehen) möglichst wenig Auslagerungen notwendig werden
- Häufig verwendete Strategien:
 - First-In-First-Out-Strategien (FIFO)
 - Not-Recently-Used-Strategie (NRU)
 - Least-Recently-Used-Strategie (LRU)
 - Least-Frequently-Used-Strategie (LFU)

Die optimale Strategie

Ersetze die Seite, die am **spätesten** in der Zukunft benötigt wird (Belady, 1966)

Referenzfolge: 0, 1, 2, 3, 1, 4, 5, 0, 2, 6, 4, 5, 6, 3, 5, ...

RAM	0	0	0	0	0	0	0	0	2	6	6	6	6
RAM		1	1	1	1	4	4	4	4	4	4	4	4
RAM			2	3	3	3	5	5	5	5	5	5	5
DISK				2	2	2	2	2	0	0	0	0	0
DISK						1	1	1	1	1	1	1	1
DISK							3	3	3	3	3	3	3
DISK									2	2	2	2	2

FIFO-Strategien

Ersetze die Seite, die **bereits am längsten** im Hauptspeicher steht.
 Abwandlung: Falls die Seite ein gesetztes Referenced-Bit besitzt, behandle sie, als wäre sie **neu eingelagert** worden (**second chance**)

Referenzfolge: 0, 1, 2, 3, 1, 4, 5, 0, 2, 6, 4, 5, 6, 3, 5, ...

RAM	0	0	0	3	3	3	3	0	0	0	4	4	4
RAM		1	1	1	1	4	4	4	2	2	2	5	5
RAM			2	2	2	2	5	5	5	6	6	6	6
DISK				0	0	0	0	3	3	3	3	3	3
DISK						1	1	1	1	1	1	1	1
DISK							2	2	4	4	0	0	0
DISK										5	5	2	2

NRU-Strategie

Ersetze eine Seite, die **in letzter Zeit nicht benutzt** wurde. Berücksichtige das Referenced- und das Modified-Bit. Ersetze in der Reihenfolge: R=0 und M=0, R=0 und M=1, R=1 und M=0, R=1 und M=1.

Referenzfolge: 0, 1, 2, 3, 1, 4, 5, 0, 2, 6, 4, 5, 6, 3, 5, ...

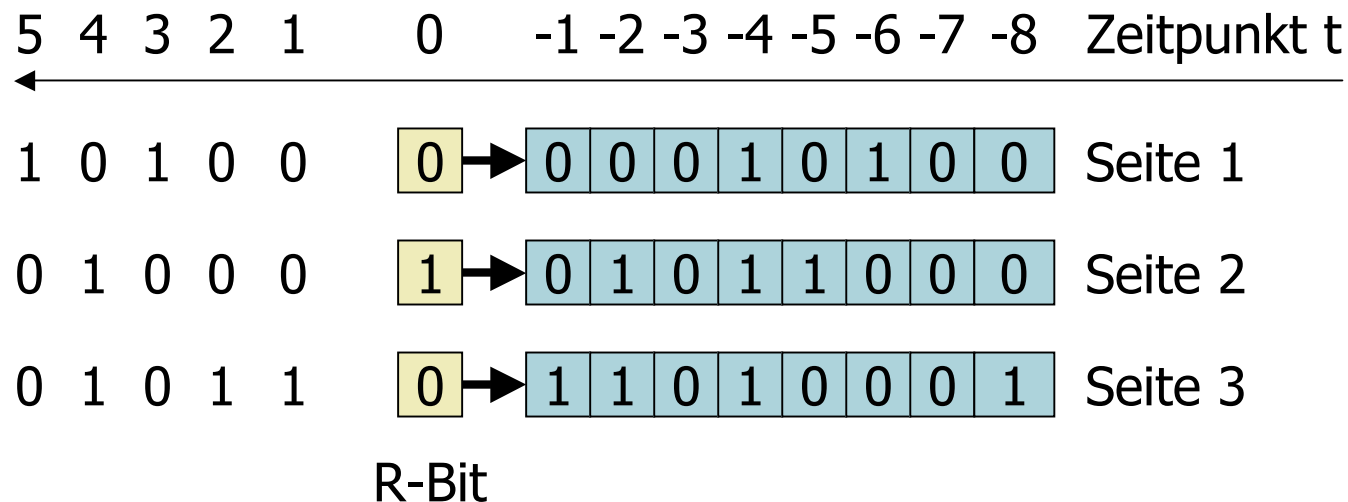
(keine Seite wird modifiziert, R wird nach 2 Takten zurückgesetzt)

RAM	0	0	0	3	3	3	5	5	5	6	6	6	6
RAM		1	1	1	1	1	1	0	0	0	4	4	4
RAM			2	2	2	4	4	4	2	2	2	5	5
DISK				0	0	0	0	1	1	1	1	1	1
DISK						2	2	2	4	4	0	0	0
DISK							3	3	3	3	3	3	3
DISK										5	5	2	2

LRU-Strategie

Ersetze die Seite, die am längsten nicht mehr benutzt wurde.

Realisierung: - Zeitstempel (ersetze älteste Seite mit R=0)
- Schieberegister (ersetze Seite mit kleinsten Wert)



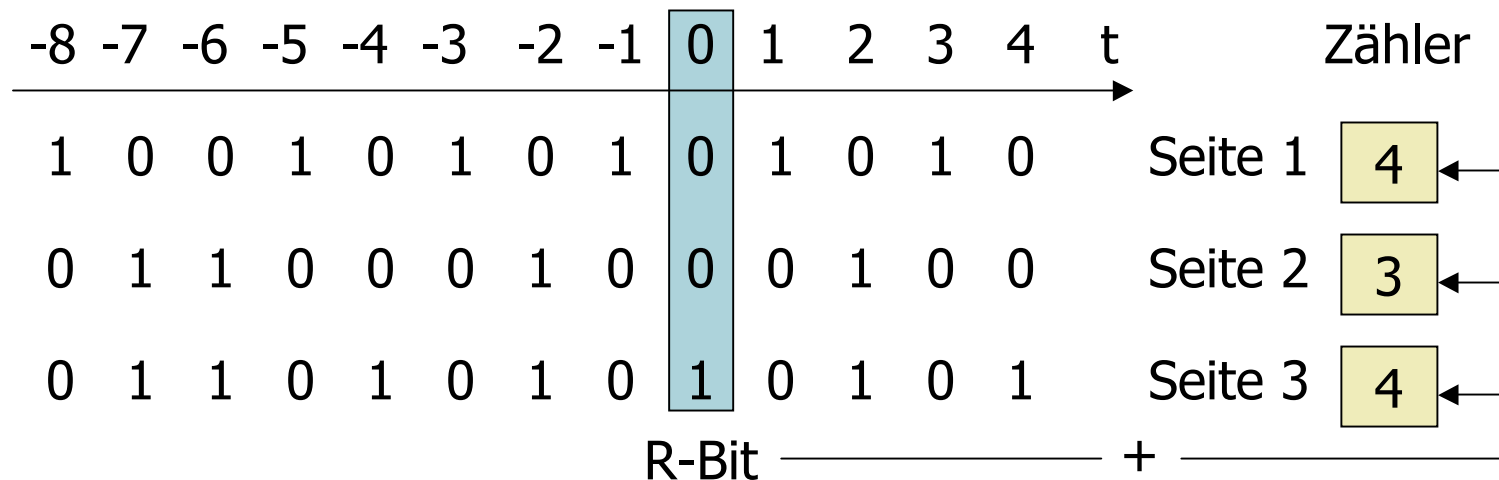
Realisierung der LRU-Strategie mit Schieberegistern

LFU-Strategie

Ersetze die Seite, die bisher **am seltensten benutzt** wurde

Realisierung: Zähler, der bei gesetztem R-Bit erhöht wird

Varianten: Einbau von **Änderungsmechanismen**, z. B. periodisches Rücksetzen des Zählers oder Verschieben des Werts nach rechts





Wichtige Begriffe

- **Arbeitsmenge (working set):**
Menge der Seiten, die im Hauptspeicher vorhanden sein müssen, damit ein Prozess (innerhalb eines bestimmten Zeitfensters) **effizient** arbeiten kann
- **Seitenfehler (page fault):**
Zugriff auf eine nicht im Hauptspeicher befindliche Seite. Ein Seitenfehler führt zur **Unterbrechung** des Prozesses und der **Aktivierung** des Betriebssystems
- **Seitenflattern (thrashing):** Das Betriebssystem muss aufgrund einer Überlastung rechenwilliger Prozesse **ständig** Seiten ein- und auslagern. Die Ausführung der Prozesse kommt praktisch zum **Stillstand**.



Vor- und Nachteile der Seitenersetzung

■ Vorteile

- Nur die **aktuell benötigten Teile** des Prozesses müssen im Hauptspeicher vorhanden sein
- einfache Freispeicherverwaltung
- keine Fragmentierung
- Überlappende Segmente möglich (**shared code**)

■ Nachteile

- Spezifische Merkmale (von Segmenten) müssen bei **allen korrespondierenden Seiten** eingetragen werden
- Seitentabellen sind **wesentlich größer** als Segmenttabellen
- **Platzverlust** durch nicht vollständig belegte Seiten (**Verschnitt**)



Aufgabe

- Gegeben sei ein System mit Seitenadressierung. Die Adresslänge beträgt 16 Bit. Für die Seitennummer werden die höchstwertigen drei Bits verwendet, die restlichen 13 Bit bilden den Offset.

Ermitteln Sie unter Verwendung der unten stehenden Tabelle die physikalischen Adressen folgender virtueller Adressen:

6815_H

2711_H

AAFE_H

Seite	Rahmen
0	4
1	2
2	7
3	0

Seite	Rahmen
4	1
5	6
6	5
7	3



Aufgabe

- Ein System besitzt drei Seitenrahmen, die zunächst nicht belegt sind. Nun werden nacheinander folgende Seiten referenziert: 0, 1, 2, 3, 4, 0, 1, 5, 2, 6, 5, 2, 6, 3, 0, 1, 3, 4, 5, 2, 6, 4, 3, 0...
- a) Zeichnen Sie für die ersten zwölf Referenzierungen ein Diagramm mit den ein- und ausgelagerten Seiten für die optimale Seitenersetzungsstrategie, die FIFO-Strategie (ohne second chance) sowie die LRU-Strategie (mit Zeitstempel).
- b) Wie viele Seitenfehler treten bei den einzelnen Strategien auf?
- c) Wie ändert sich die Anzahl der Seitenfehler bei den einzelnen Strategien, wenn dem System ein vierter Seitenrahmen hinzugefügt wird?



Aufgabe

- Ein Computersystem besitzt vier Seitenrahmen. Für jeden Rahmen werden die Seitennummer, der Zeitpunkt des Ladens, der Zeitpunkt des letzten Zugriffs sowie die Referenced- und Modified-Bits vermerkt. Folgende Tabelle gibt den gegenwärtigen Zustand wieder:

Seite	geladen	letzter Zugriff	R-Bit	M-Bit
0	356	397	1	0
1	128	412	1	1
2	218	372	0	0
3	189	210	0	1

- Wenn eine Seite ausgelagert werden muss, welche Seite wird nach der FIFO-, Second-Chance-, NRU- und LRU-Strategie gewählt?