



Betriebssysteme

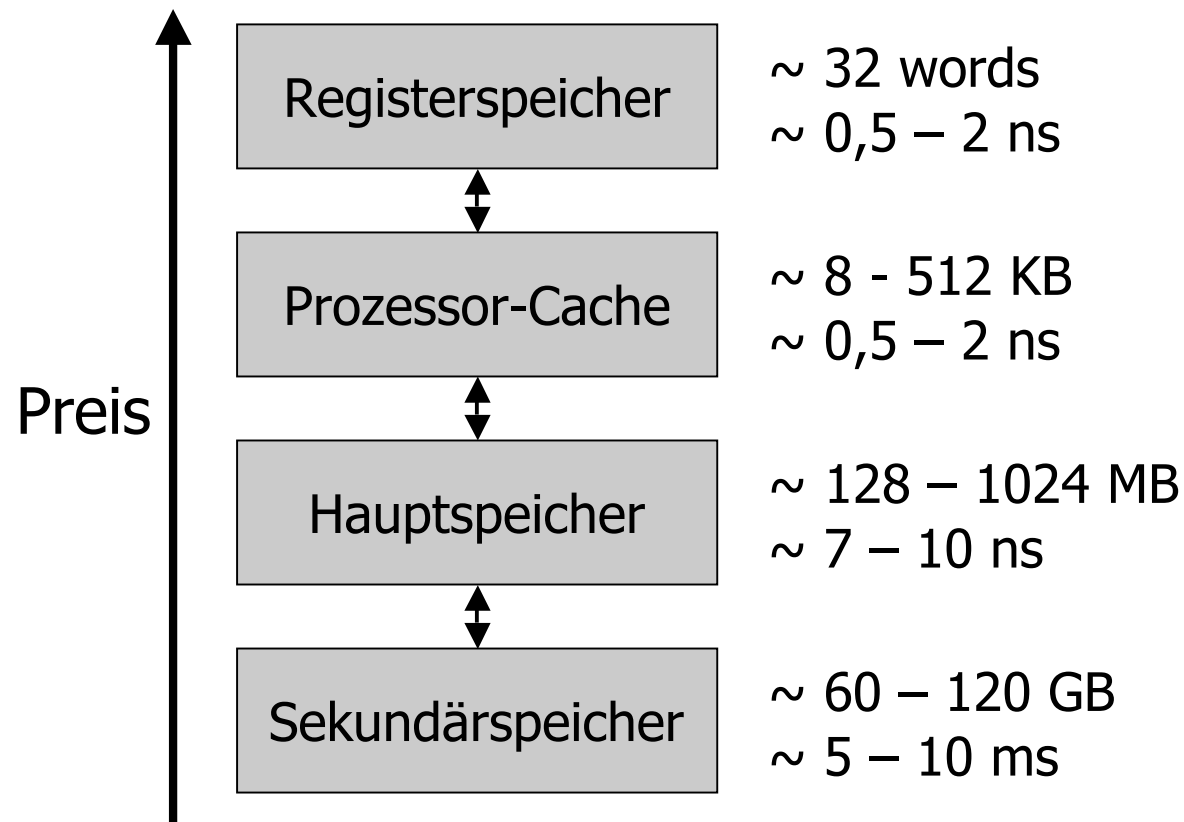
Speicherverwaltung I: Virtueller Speicher, Segmentierung



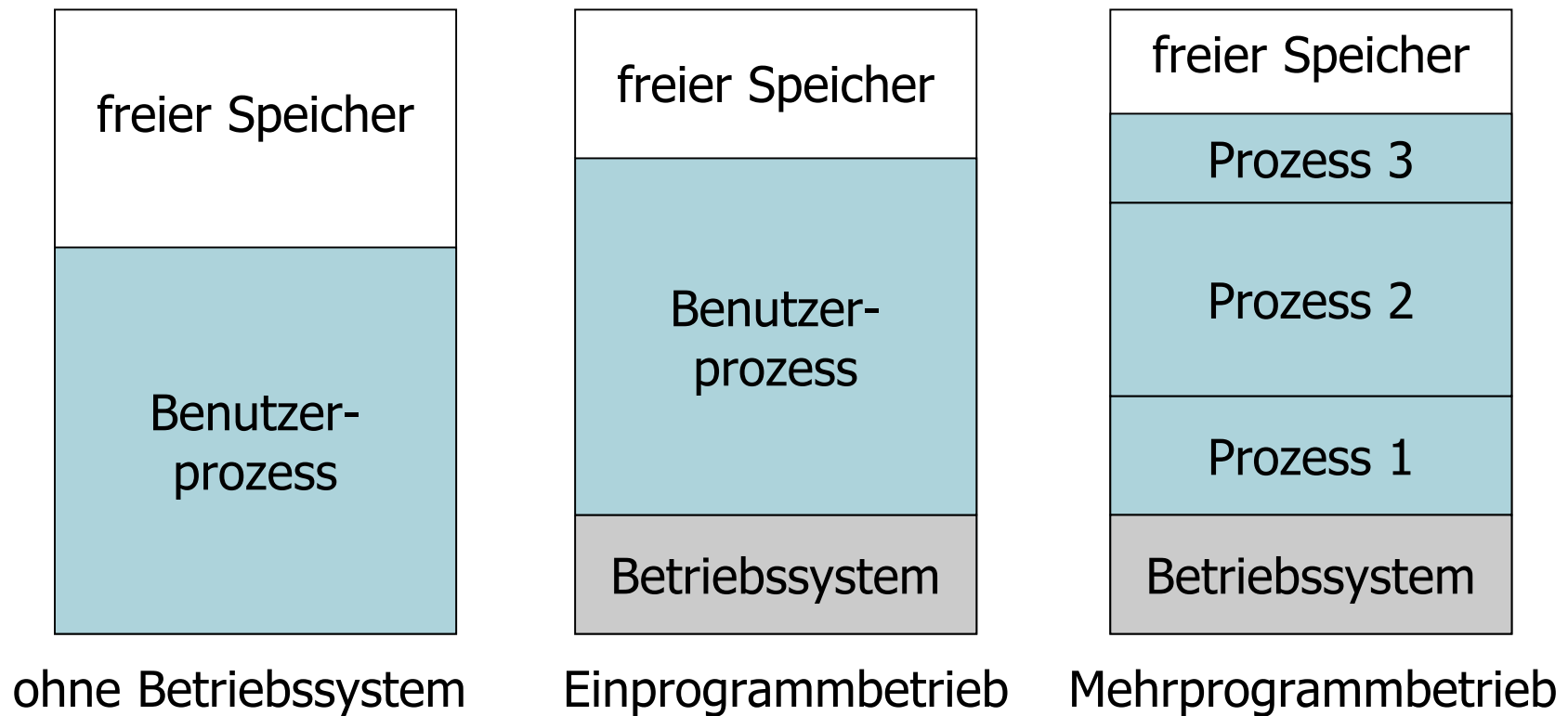
Überblick

- Speichernutzungstechniken
- Virtueller Speicher
- Segmentierung
- Einlagern benötigter Segmente
- Belegungsstrategien
- Vor- und Nachteile der Segmentierung

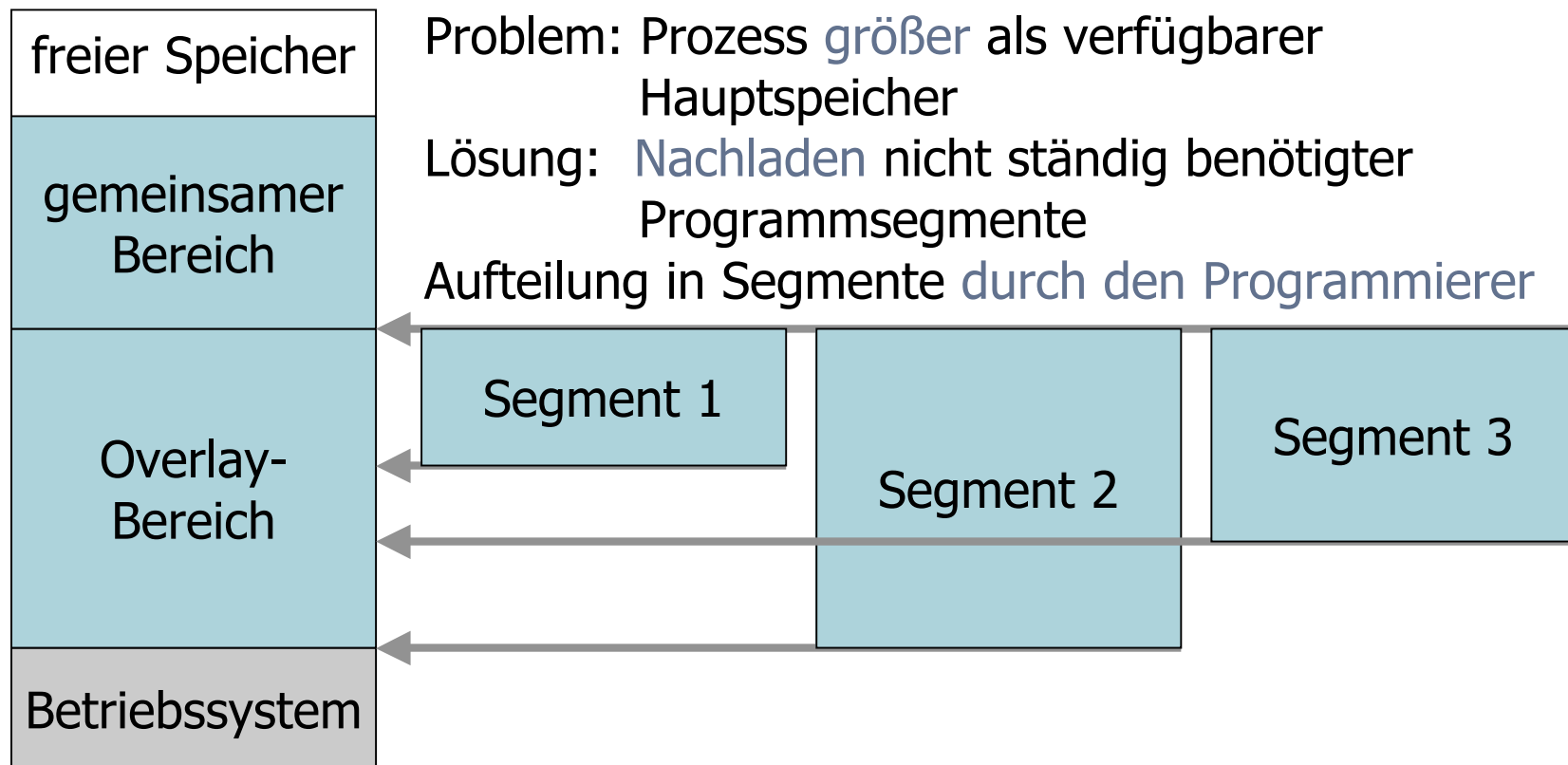
Speicherhierarchie



Aufteilung des Hauptspeichers



Overlay-Technik





Verschiebung des Programmcodes (relocation)

- Problem: Absolute Adressierung nur möglich, wenn die Position des Programms im Hauptspeicher bekannt ist
- Lösungsmöglichkeiten:
 - anpassen der absoluten Adressen beim Laden des Programms (relocating loader)
 - ermitteln der effektiven Adressen bei Ausführung des Befehls (Voraussetzung: spezielles Adressrechenwerk im Prozessor, z. B. Basisadressregister)



Virtueller Speicher

- Jeder Prozess erhält einen eigenen virtuellen (logischen) Adressraum
- Der virtuelle Speicher wird auf den Sekundärspeicher abgebildet
- Programm- und Datenbereiche sind nicht durch die Größe des realen Hauptspeichers begrenzt
- Ausführung von mehreren Programmen, deren Gesamtgröße die Größe des realen Hauptspeichers überschreiten, ist möglich



Virtuelle Speicherverwaltung

- Der reale/physikalische Hauptspeicher enthält in der Regel nur die augenblicklich benötigten Programmabschnitte und Daten
- Zur Zeit nicht benötigte Programmabschnitte und Daten können durch das Betriebssystem auf den Sekundärspeicher ausgelagert werden
- Erst im Mehrprogrammbetrieb effizient nutzbar.



Mechanismen der virtuellen Speicherverwaltung

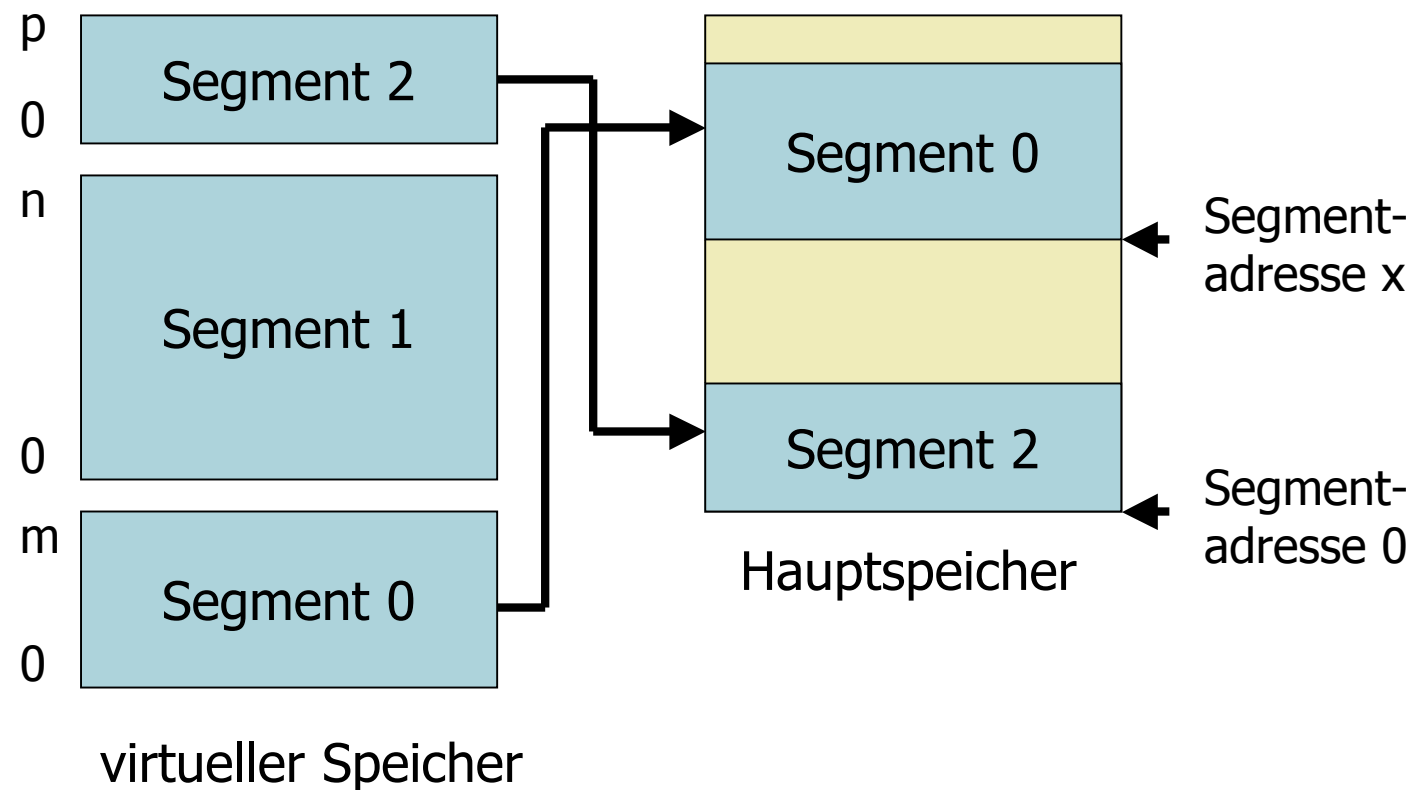
- Hardware-Unterstützung durch Memory-Management-Unit (MMU)
- Speicherschutz:
 - Isolierung der Adressräume
 - Überprüfung der Zugriffsarten
- Grundlegende Abbildungsmechanismen:
 - Segmentierung
 - Seitenadressierung (paging)



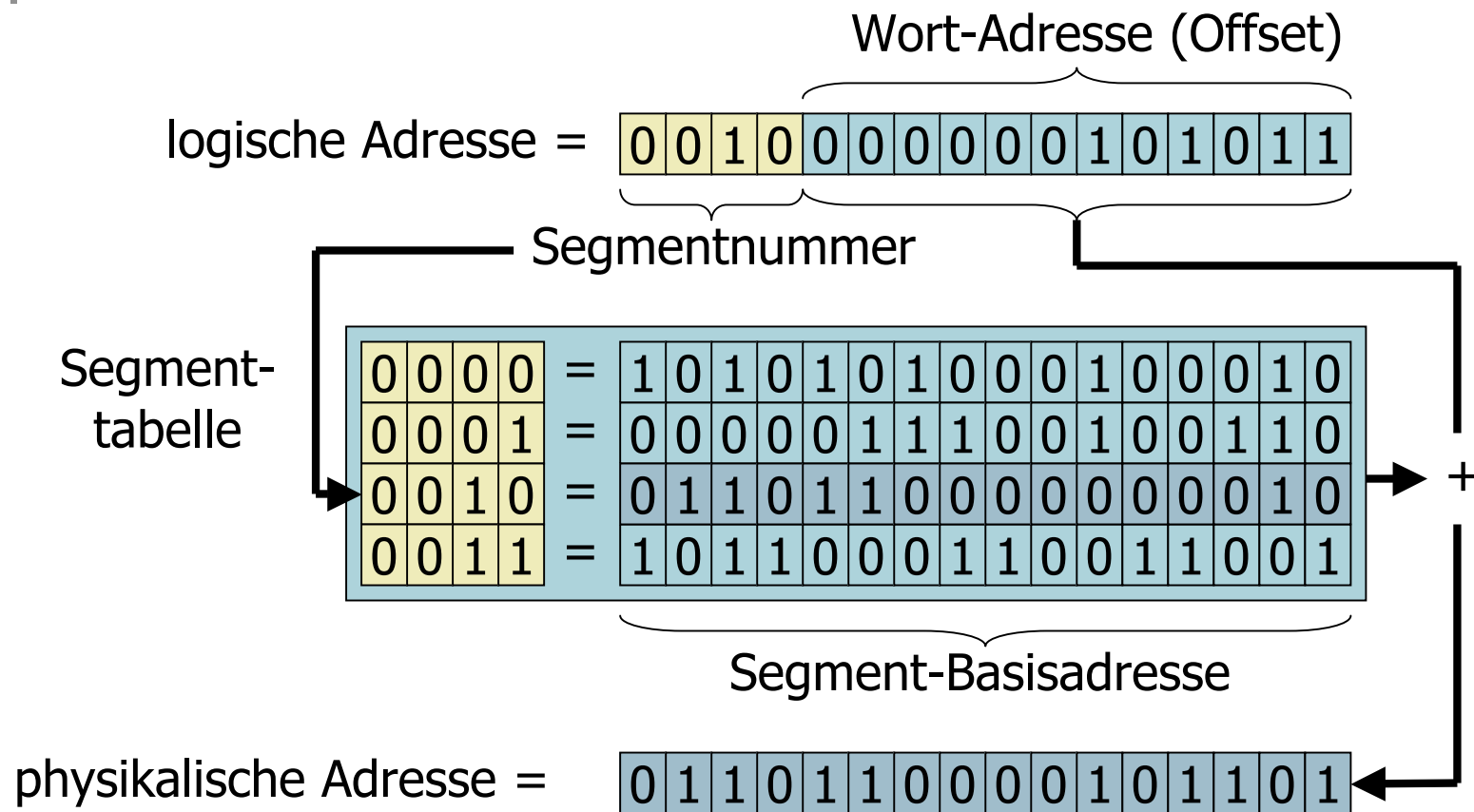
Segmentierung

- Unterteilung des logischen Adressraums in Segmente:
 - Abschnitte **variabler** Größe
 - Aufteilung gemäß den **zusammengehörenden Einheiten** des Programms (Unterprogramme, Datenbereiche usw.)
 - typische Größe: 256 Byte – 64 KB

Abbildung der Segmente auf den Hauptspeicher



Logische und reale Adressen bei Segmentierung





Segmenttabellen

- Tabelle mit Informationen zu allen Segmenten eines Prozesses:
 - Segmentnummer
 - Segment-Basisadresse
 - Längenangabe (für Bereichsschutz)
 - Zugriffsattribute (für Zugriffsschutz)
 - Markierung geladen/nicht geladen
 - Markierung verändert/nicht verändert (dirty tag)

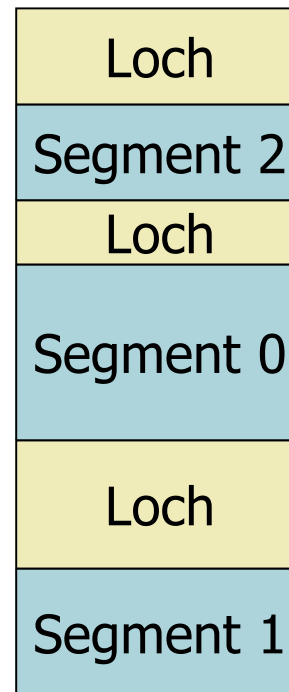
Einlagern benötigter Segmente: Fall 1

Mindestens eine Lücke ausreichender Größe im Hauptspeicher vorhanden

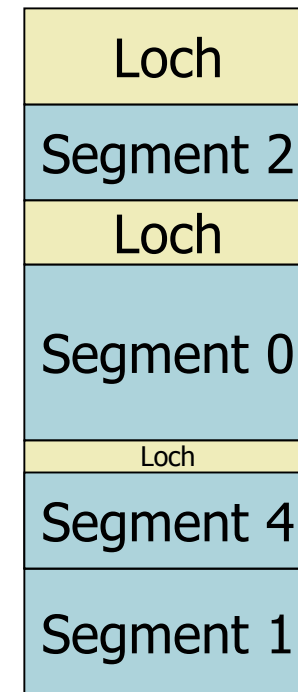
Segment 4



Vorgehen: geeignete Lücke für das Segment auswählen



Hauptspeicher



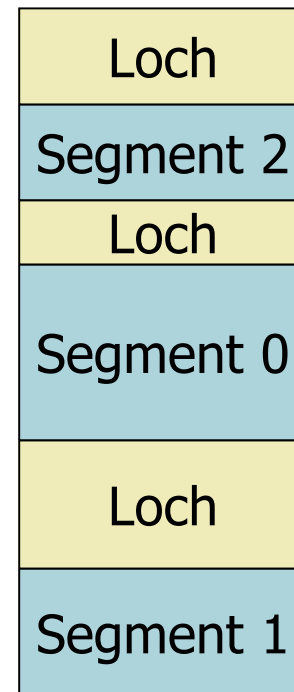
Hauptspeicher

Einlagern benötigter Segmente: Fall 2

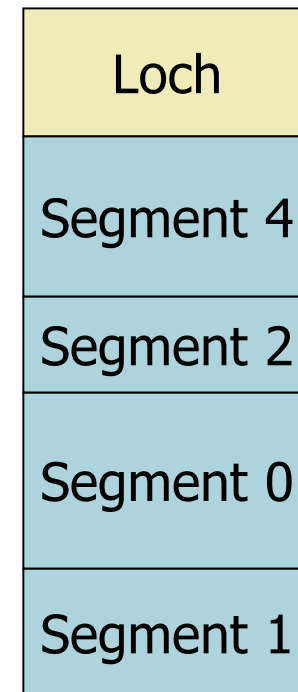
Keine Lücke ist groß genug, die Summe mehrerer Lücken würde aber ausreichen

Segment 4

Vorgehen: Platz schaffen durch Verschieben der Segmente



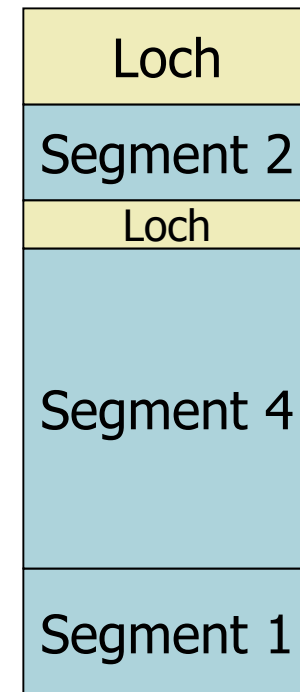
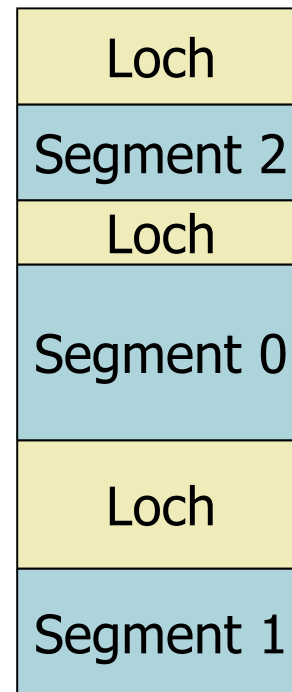
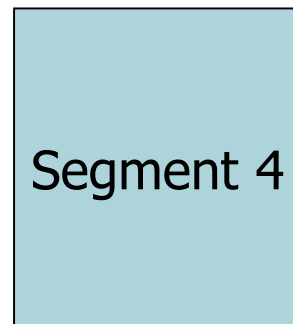
Hauptspeicher



Hauptspeicher

Einlagern benötigter Segmente: Fall 3

Der insgesamt zur
Verfügung stehende freie
Platz reicht nicht aus



Vorgehen: gerade nicht
benötigte Segmente auf
Sekundärspeicher auslagern

Hauptspeicher

Hauptspeicher



Belegungsstrategien

- **First-Fit:** Wähle die **erste**, ausreichend große Lücke
 - schnell, aber wiederholtes Durchsuchen der Restlöcher
- **Next-Fit:** Beginne bei der zuletzt gefüllten Lücke und wähle die **nächste**, ausreichend große Lücke
 - schnell, wiederholtes Durchsuchen der Restlöcher nicht notwendig
- **Best-Fit:** Wähle die **kleinste**, noch ausreichend große Lücke
 - langsamer, erhält große Restlöcher, erzeugt kleine Restlöcher
- **Worst-Fit:** Wähle die **größte** vorhandene Lücke
 - langsamer, vermeidet kleine Restlöcher

Belegungsbeispiel

Einlagerung von Segmenten der folgenden Größe:

10

9

7

vorher:

12	...	8	...	4	...	11	...	6	...	5	...	14	...	9
----	-----	---	-----	---	-----	----	-----	---	-----	---	-----	----	-----	---

First-Fit

10	2	...	7	1	...	4	...	9	2	...	6	...	5	...	14	...	9
----	---	-----	---	---	-----	---	-----	---	---	-----	---	-----	---	-----	----	-----	---

Next-Fit

10	2	...	8	...	4	...	9	2	...	6	...	5	...	7	7	...	9
----	---	-----	---	-----	---	-----	---	---	-----	---	-----	---	-----	---	---	-----	---

Best-Fit

12	...	7	1	...	4	...	10	1	...	6	...	5	...	14	...	9
----	-----	---	---	-----	---	-----	----	---	-----	---	-----	---	-----	----	-----	---

Worst-Fit

9	3	...	8	...	4	...	7	4	...	6	...	5	...	10	4	...	9
---	---	-----	---	-----	---	-----	---	---	-----	---	-----	---	-----	----	---	-----	---



Vor- und Nachteile der Segmentierung

■ Vorteile

- Segmente entsprechen zusammengehörigen Einheiten, denen spezifische Merkmale zugeordnet werden können
- Überlappende Segmente möglich (shared code)
- Segmentlänge dynamisch änderbar

■ Nachteile

- Aufgrund der variablen Segmentlänge aufwendige Freispeicherverwaltung notwendig
- Segmente müssen immer vollständig ein- oder ausgelagert werden
- Speicherzersplitterung (Fragmentierung) möglich

Aufgabe

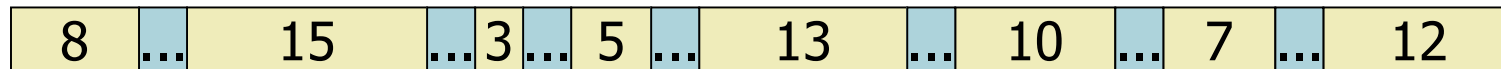
Gegeben sei der folgende Ausschnitt aus einer Segment-Tabelle (Segmentnummer 4 Bit, Offset 12 Bit):

Nummer				Basisadresse											
0	0	0	0	=	1	0	0	0	1	0	0	1	1	0	0
0	0	0	1	=	0	0	0	0	0	1	1	1	0	0	0
0	0	1	0	=	0	1	0	0	1	0	1	0	0	0	1
0	0	1	1	=	1	0	1	1	0	0	0	1	1	0	0

Ermitteln Sie für die folgenden logischen Adressen die zugehörigen physikalischen Adressen: $1F3A_H$, 3580_H , $002A_H$

Aufgabe

Der Hauptspeicher eines Computers ist zu einem bestimmten Zeitpunkt so mit sieben Segmenten belegt, dass sich acht Lücken der Größe 8, 15, 3, 5, 13, 10, 7 und 12 KB ergeben (in der angegebenen Reihenfolge).



Nun müssen nacheinander vier Segmente der Größe 4, 10, 11 und 4 KB eingelagert werden.



Zeichnen Sie für die Belegungsstrategien First-Fit, Next-Fit, Best-Fit und Worst-Fit jeweils die sich ergebende Speicherbelegung.