



universität
uulm



Betriebssysteme SS2024

Speicherverwaltung

Tutorium 09

Inhaltsverzeichnis

1 Datenstrukturen im Speicher

2 Speichervergabe

3 Adressräume

Auswahl an dynamischen Datenstrukturen (1)

■ Array

- Besteht aus einem fest definierten Speicherbereich.
- Benötigt exakt so viele aufeinanderfolgende Speicherblöcke wie Datenblöcke.
- Die Größe des Arrays wird (hier) nicht mit abgelegt → Arrays in C sind so realisiert

■ Verkettete Listen

- Der jeweils erste Block speichert die Daten
- Der zweite Block enthält eine Referenz auf den ersten Block des nächsten Elements
- Eine Referenz auf die Adresse FF_{16} markiert (hier) das Ende der Liste

■ Verkettete Array-Listen

- Verkettung von Array-Blöcken mit variabler Größe
- Jeder der Array-Blöcke besteht (hier) aus:
 - einem initialen Speicherblock der die Größe des (gesamten) Array-Blocks angibt
 - einem weiteren Speicherblock mit einer Referenz auf den nächsten Array-Block
 - einer beliebigen Anzahl an Datenblöcken
- Eine Referenz auf die Adresse FF_{16} markiert (hier) wieder das Ende der Array-Liste

Auswahl an dynamischen Datenstrukturen (2)

■ Array

- Besteht aus einem fest definierten Speicherbereich.
- Benötigt exakt so viele aufeinanderfolgende Speicherblöcke wie Datenblöcke.
- Die Größe des Arrays wird (hier) nicht mit abgelegt → Arrays in C sind so realisiert

■ Verkettete Listen

- Der jeweils erste Block speichert die Daten
- Der zweite Block enthält eine Referenz auf den ersten Block des nächsten Elements
- Eine Referenz auf die Adresse FF_{16} markiert (hier) das Ende der Liste

■ Verkettete Array-Listen

- Verkettung von Array-Blöcken mit variabler Größe
- Jeder der Array-Blöcke besteht (hier) aus:
 - einem initialen Speicherblock der die Größe des (gesamten) Array-Blocks angibt
 - einem weiteren Speicherblock mit einer Referenz auf den nächsten Array-Block
 - einer beliebigen Anzahl an Datenblöcken
- Eine Referenz auf die Adresse FF_{16} markiert (hier) wieder das Ende der Array-Liste

Auswahl an dynamischen Datenstrukturen (3)

■ Array

- Besteht aus einem fest definierten Speicherbereich.
- Benötigt exakt so viele aufeinanderfolgende Speicherblöcke wie Datenblöcke.
- Die Größe des Arrays wird (hier) nicht mit abgelegt → Arrays in C sind so realisiert

■ Verkettete Listen

- Der jeweils erste Block speichert die Daten
- Der zweite Block enthält eine Referenz auf den ersten Block des nächsten Elements
- Eine Referenz auf die Adresse FF_{16} markiert (hier) das Ende der Liste

■ Verkettete Array-Listen

- Verkettung von Array-Blöcken mit variabler Größe
- Jeder der Array-Blöcke besteht (hier) aus:
 - einem initialen Speicherblock der die Größe des (gesamten) Array-Blocks angibt
 - einem weiteren Speicherblock mit einer Referenz auf den nächsten Array-Block
 - einer beliebigen Anzahl an Datenblöcken
- Eine Referenz auf die Adresse FF_{16} markiert (hier) wieder das Ende der Array-Liste

Beispiel

Speichern Sie die Werte 51, 61, 71, und 81 in dieser Reihenfolgen mit den zuvor genannten Verfahren jeweils in einer Kopie des folgenden Speicherabschnitts:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten					56	0D							57	FF		

Beispiel - Lösung

Speichern Sie die Werte 51, 61, 71, und 81 in dieser Reihenfolgen mit den zuvor genannten Verfahren jeweils in einer Kopie des folgenden Speicherabschnitts:

Array:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	51	61	71	81	56	0D							57	FF		

Beispiel - Lösung

Speichern Sie die Werte 51, 61, 71, und 81 in dieser Reihenfolgen mit den zuvor genannten Verfahren jeweils in einer Kopie des folgenden Speicherabschnitts:

Array:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	51	61	71	81	56	0D							57	FF		

Verkettete Liste:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	51	02	61	06	56	0D	71	08	81	FF			57	FF		

Beispiel - Lösung

Speichern Sie die Werte 51, 61, 71, und 81 in dieser Reihenfolgen mit den zuvor genannten Verfahren jeweils in einer Kopie des folgenden Speicherabschnitts:

Array:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	51	61	71	81	56	0D							57	FF		

Verkettete Liste:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	51	02	61	06	56	0D	71	08	81	FF			57	FF		

Verkettete Array-Liste:

Adresse	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Daten	04	06	51	61	56	0D	04	FF	71	81			57	FF		

Speichervergabe Strategien

- Ein Segment ist ein zusammenhängender Speicherbereich mit aufeinanderfolgenden Adressen
- Segmente können unterschiedliche Längen haben
- Ein Anwendungsprogramm besitzt mindestens die folgende Segmente
 - Code-Segment: Instruktionen des Programms
 - Daten-Segment: Daten des Programms
 - Stack-Segment: Parameter für Funktionsaufrufe

Speichervergabe - First Fit Strategie

- Die Suche nach einem freien Segment beginnt immer am Anfang des Speicherbereichs
- Das erste passende freie Segment wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.
Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke



Speichervergabe - First Fit Strategie

- Die Suche nach einem freien Segment beginnt immer am Anfang des Speicherbereichs
- Das erste passende freie Segment wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.
Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke

Lösung:



Speichervergabe - Next Fit Strategie

- Die Suche nach einem freien Segment beginnt beim Segment welches zuletzt belegt wurde
- Das erste passende freie Segment wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke



Speichervergabe - Next Fit Strategie

- Die Suche nach einem freien Segment beginnt beim Segment welches zuletzt belegt wurde
- Das erste passende freie Segment wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke

Lösung:



Speichervergabe - Best Fit Strategie

- Die am besten passende Lücke wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke



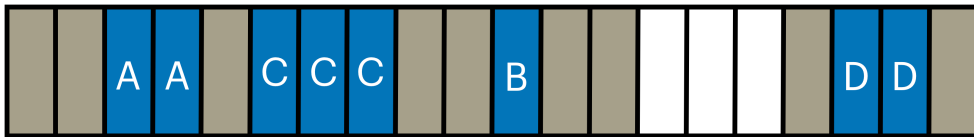
Speichervergabe - Best Fit Strategie

- Die am besten passende Lücke wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke

Lösung:



Speichervergabe - Worst Fit Strategie

- Die am schlechtesten passende Lücke wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke



Speichervergabe - Worst Fit Strategie

- Die am schlechtesten passende Lücke wird ausgewählt

Aufgabe: Platzieren Sie folgende Dateien in dem gegebenen Speicherbereich.

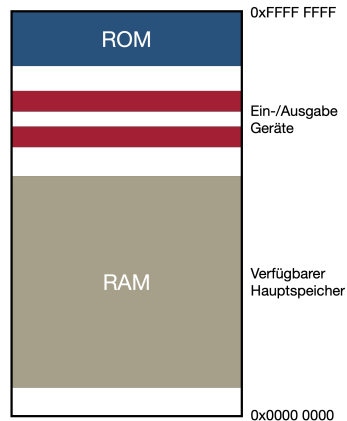
Datei A: 2 Blöcke, Datei B: 1 Block, Datei C: 3 Blöcke, Datei D: 2 Blöcke

Lösung: Datei C kann nicht gespeichert werden



Physikalischer Adressraum

- Darstellung des Adressraums einer 32 Bit Architektur
- Physikalische Adressen sind von 0 bis $2^{32}-1$ verfügbar
- Nicht alle Adressen sind auch auf Hardware-Ebene verfügbar



Logischer Adressraum

- Wird alternativ auch als virtueller Adressraum bezeichnet
- Jeder Prozess hat einen eigenen logischen Adressraum mit 2^{32} Adressen
- Erlaubt Programmen beliebige Adressen zu verwenden, ohne Kollisionen im Mehrprogrammbetrieb zu verursachen
- Erfordert eine Übersetzung in den physikalischen Adressraum
- Schützt zusätzlich vor nicht-erlaubten Zugriffen in Speicherbereichen anderer Programme

Ausblick: Realisierung der Übersetzung

