

Einführung in die Informatik: Systeme und Anwendungen
SS 2014

Übungsblatt 5: Speicherverwaltung

Abgabe bis 19.05.2014, 14:00 Uhr

Besprechung am 19.05./20.05.2014

Aufgabe 5-1 *Speicherbelegungsstrategien*

Hausaufgabe

(2+2+2+1 Punkte)

Eine Frei-Liste beschreibt die ungenutzten Bereiche innerhalb eines Speichers. Sie wird durch eine Folge von Zahlen dargestellt, welche die Größe der ungenutzten Bereiche in der Reihenfolge ihres Vorkommens im Speicher repräsentiert. Hier wird die genaue Position dieser Bereiche außer acht gelassen.

Betrachten Sie die Frei-Liste [10, 21, 14, 18, 7]. Welche Frei-Listen entstehen nacheinander bei Verwendung des Verfahrens

(a) First Fit

(b) Next Fit

(c) Best Fit

wenn vom Betriebssystem Speicherbereiche der Größe 18, 9, 6, 12 und 8 in dieser Reihenfolge angefordert werden?

Zur Erklärung:

- First Fit sucht stets vom Anfang der Frei-Liste aus den ersten Bereich, der groß genug ist, um die Anforderung zu erfüllen.
- Next Fit arbeitet so wie First Fit mit dem Unterschied, dass nicht jedes Mal am Anfang der Frei-Liste begonnen wird, nach dem ersten passenden Platz zu suchen, sondern an der Stelle, an der zuletzt ein freier Platz belegt wurde. Ist das Ende der Frei-Liste erreicht, und kein freier Platz wurde gefunden, so wird die Suche am Anfang fortgesetzt. Bei der ersten Anforderung wird am Anfang der Liste mit der Suche begonnen.
- Best Fit durchsucht stets die gesamte Frei-Liste nach dem kleinsten Bereich, der die Anforderung erfüllt.

(d) Welches Verfahren scheint Ihnen allgemein das beste zu sein?

Aufgabe 5-2 *Speicherbelegungsstrategien*
Hausaufgabe

(3+3+3 Punkte)

In dieser Aufgabe sollen die drei Speicherbelegungsverfahren First Fit, Best Fit und Next Fit miteinander verglichen werden.

Die Güte eines solchen Verfahrens kann z.B. durch Vergleiche der Frei-Listen, die diese Verfahren auf eine bestimmte Situation und eine Folge von Anforderungen durch das Betriebssystem herstellen, bestimmt werden.

Offensichtlich erzeugen verschiedene Verfahren angewandt auf dieselbe Frei-Liste zu Anfang und dieselbe Folge von Anforderungen auch dieselbe Größe freien Speichers – vorausgesetzt, dass jede Anforderung auch bedient werden kann. Das bedeutet, dass die Summe über alle Elemente der entstehenden Frei-Listen in allen Fällen gleich ist.

Der freie Speicher kann jedoch am Ende in verschieden große Stücke aufgeteilt sein. Wir nennen an dieser Stelle ein Verfahren F_1 *schlechter* als F_2 bezüglich einer gegebenen Frei-Liste und einer Folge von Anforderungen, wenn die Frei-Liste, die F_1 erzeugt, länger ist als diejenige, die F_2 erzeugt. In diesem Fall hat F_1 also den freien Speicher in mehr Stücke aufgeteilt als F_2 .

Finden Sie für jeden der folgenden Fälle (a), (b) und (c) eine anfängliche Frei-Liste und eine Folge von Anforderungen, so dass bei allen Verfahren alle Anforderungen bedient werden können und bezüglich dieser

- (a) First Fit schlechter ist als Next Fit und Best Fit,
- (b) Next Fit schlechter ist als First Fit und Best Fit,
- (c) Best Fit schlechter ist als First Fit und Next Fit.

Geben Sie bei jeder Aufgabe jeweils folgendes an:

- die anfängliche Frei-Liste,
- die Liste der Anforderungen,
- jeweils die Frei-Liste, die durch First Fit, Next Fit bzw. Best Fit nach Abarbeitung aller Anforderungen entstanden ist.

Um die Korrektur dieser Aufgabe zu erleichtern, erstellen Sie bitte für die Lösung eine Tabelle gemäß der Vorlage, die wir Ihnen auf der Website zum Download bereitstellen.

Aufgabe 5-3 *Seitenersetzungsstrategien*

(3+3+3 Punkte)

Beim Paging werden angeforderte Seiten nach Bedarf in freie Seitenrahmen des Hauptspeichers eingelagert. Eine priorisierte Seitenliste ist eine endliche Zahlenfolge und enthält die Nummern der sich momentan im Hauptspeicher befindlichen Seiten. Die Seiten sind in der Liste nach Priorität geordnet. Ganz rechts steht die Seite mit der niedrigsten Priorität, d.h. diejenige Seite, die beim nächsten Seitenfehler in den Hintergrundspeicher ausgelagert wird. Es gibt eine Reihe verschiedener Seitenersetzungsstrategien, z.B.:

- Die *optimale Strategie* (OPT) ersetzt die Seite, die in Zukunft am längsten nicht angefordert werden wird.
- *First In First Out* (FIFO) ersetzt die Seite, die sich am längsten im Hauptspeicher befindet.
- *Least Recently Used* (LRU) ersetzt die Seite, die am längsten nicht mehr angefordert wurde.

Die optimale Strategie kann in der Praxis nicht verwendet werden, weil zukünftige Seitenzugriffe nicht im Voraus bekannt sind. Sie ist jedoch ein guter Maßstab zum Vergleich mit anderen Strategien.

Wir betrachten einen Hauptspeicher, der in drei Seitenrahmen unterteilt ist. Welche priorisierten Seitenlisten entstehen, wenn vom Betriebssystem die Seiten

2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

in dieser Reihenfolge bei Verwendung der Verfahren

- (a) OPT
(b) FIFO
(c) LRU

angefordert werden, wenn sich am Anfang keine Seite im Hauptspeicher befindet? Wieviele Seitenfehler ergeben sich bei den verschiedenen Verfahren? Geben Sie jeweils den Status der priorisierten Seitenlisten nach jeder einzelnen Anforderung an.

Aufgabe 5-4 *Seitenersetzungsstrategien*

Hausaufgabe

(3+3+3+3+1 Punkte)

Welche priorisierten Seitenlisten entstehen, wenn vom Betriebssystem die Seiten

$$0, 1, 2, 3, 0, 1, 4, 0, 1, 2, 3, 4$$

in dieser Reihenfolge bei Verwendung der Verfahren

- (a) First In First Out (FIFO) mit drei Seitenrahmen
- (b) First In First Out (FIFO) mit vier Seitenrahmen
- (c) Least Recently Used (LRU) mit drei Seitenrahmen
- (d) Least Recently Used (LRU) mit vier Seitenrahmen

angefordert werden, wenn sich am Anfang keine Seite im Hauptspeicher befindet?

Geben Sie jeweils den Status der priorisierten Seitenlisten nach jeder einzelnen Anforderung an. Für die Seitenersatzungsstrategien sowie die Notation gelten dieselben Regeln wie in Aufgabe 5-3.

- (e) Wie entwickelt sich die Anzahl der Seitenfehler bei FIFO in Teilaufgabe (b) im Vergleich zu (a)? Wie entwickelt sich die Anzahl der Seitenfehler bei LRU in Teilaufgabe (d) im Vergleich zu (c)? Welches der beiden Verfahren erscheint aufgrund dieser Beobachtung als das bessere? Begründen Sie kurz.