

Lösungsvorschläge – Blatt 10

Zürich, 16. Mai 2024

Lösung zu Aufgabe 12

Wir betrachten also FIFO für zwei Cache-Grössen 3 und 4. Wie immer nehmen wir an, dass die Caches mit (1, 2, 3) bzw. (1, 2, 3, 4) initialisiert worden sind. Sei nun die Eingabe

(5, 6, 7, 8, 5, 6, 9, 5, 6, 7, 8, 9)

gegeben. Mit einer Cache-Grösse von 3 ergibt sich Folgendes. Unabhängig davon, wie die ersten drei Seiten gelöscht werden, hat FIFO nach den ersten drei Anfragen den Cache-Inhalt

5	6	7
---	---	---

.

Bislang gab es also drei Seitenfehler. Während der nächsten vier Anfragen 8, 5, 6, 9 ist der Cache-Inhalt jeweils

8	6	7
---	---	---

,

8	5	7
---	---	---

,

8	5	6
---	---	---

,

9	5	6
---	---	---

und es kommt zu einem Seitenfehler in jedem Zeitschritt. Die nächsten beiden Anfragen 5 und 6 führen wiederum zu keinem Seitenfehler. Schliesslich werden 7 und 8 angefragt und der Cache-Inhalt ist

9	7	6
---	---	---

 und

9	7	8
---	---	---

,

sodass die letzte Anfrage 9 keinen Seitenfehler verursacht. Insgesamt ergeben sich somit also neun Seitenfehler.

Jetzt betrachten wir FIFO mit einer Cache-Grösse von 4 auf derselben Instanz. Nach den ersten vier Anfragen ist der Cache-Inhalt

5	6	7	8
---	---	---	---

und es gab bis hierhin vier Seitenfehler. Die nächsten zwei Anfragen 5 und 6 verursachen keine Seitenfehler. Die darauf folgenden Anfragen 9, 5, 6, 7, 8, 9 führen allerdings jeweils zu einem Seitenfehler und Cache-Inhalt ist währenddessen

9	6	7	8
---	---	---	---

,

9	5	7	8
---	---	---	---

,

9	5	6	8
---	---	---	---

,

9	5	6	7
---	---	---	---

,

8	5	6	7
---	---	---	---

,

8	9	6	7
---	---	---	---

.

Damit kommen wir insgesamt auf zehn Seitenfehler, obwohl der Cache grösser ist.

Lösung zu Aufgabe 13

Der Definition von Paging aus der Vorlesung folgend nehmen wir wieder an, dass der Cache jedes Algorithmus mit $(1, \dots, k)$ initialisiert ist.

- (a) Für ein beliebiges n betrachten wir die Eingabe $(k+1, k, k+1, k, \dots)$ der Länge n . Es ist offensichtlich, dass MAX einen Seitenfehler in jedem Zeitschritt verursacht. Auf der anderen Seite existiert ein optimaler Offline-Algorithmus OPT, der im ersten Zeitschritt beispielsweise die Seite mit dem Index 1 verdrängt und hiernach keine weiteren Kosten verursacht. Somit ist der kompetitive Faktor von MAX auf dieser Eingabe n .
- (b) Analog zu (a) betrachten wir für ein beliebiges n die Eingabe $(k+1, 1, 2, 1, 2, \dots)$ der Länge n . Die Behauptung folgt sofort.
- (c) In diesem Fall funktioniert wieder eine analoge Argumentation wie bei den vorangegangenen Aufgaben mit der Eingabe $(k+1, 1, k+1, 1, \dots)$.