

Systemarchitektur SS 2021

Aufgabenblatt 11

Sie können Ihre Lösungen bis **Mittwoch, dem 07.07.2021, um 10:00 Uhr** im CMS abgeben. Geben Sie auf Ihrer Lösung Ihr Tutorium sowie die Namen und Matrikelnummern aller Gruppenmitglieder an.

Aufgabe 11.1: Pseudo-LRU

Zur Realisierung der LRU-Ersetzungsstrategie für einen k-fach vollassoziativen Cache sind mindestens $\log(k!)$ Bits nötig um die Reihenfolge der Cacheelemente bezüglich ihres letzten Zugriffs zu speichern. In der Praxis wird daher häufig auf Pseudo-LRU zurückgegriffen, das sich ähnlich zu LRU verhält, aber lediglich k-1 Bits benötigt.

Zur Veranschaulichung der Funktionsweise der Pseudo-LRU-Strategie nehmen wir an, dass diese Bits, wie in Abbildung 1 dargestellt, baumartig angeordnet sind. Hierdurch wird ein Pfad impliziert, der zum nächsten zu ersetzenden Cacheelement führt. Eine 1 (0) bedeutet hierbei, dass der rechte (linke) Zweig genommen wird.

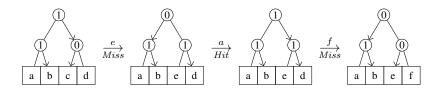


Abbildung 1: Verhalten eines 4-fach vollassoziativen PLRU Caches für die Zugriffsequenz e, a, f.

Nach einem Zugriff (sowohl Hit als auch Miss) werden die Bits auf dem Zugriffspfad so angepasst, dass sie jeweils vom zugegriffenen Element wegzeigen. In Abbildung 1 sind insgesamt drei solcher Zugriffe dargestellt. Ausgehend von dem rechten PLRU-Cachezustand in Abbildung 1 soll die Zugriffsequenz

adbcdefead

abgearbeitet werden. Geben Sie die jeweiligen Zwischenschritte (Cachezustände) explizit an.

Aufgabe 11.2: Ersetzungsstrategien im Vergleich

Gegeben ein leerer, 4-fach vollassoziativer Cache. Geben Sie jeweils eine Zugriffssequenz an, so dass

- (a) LRU weniger Misses verursacht als FIFO
- (b) FIFO weniger Misses verursacht als LRU
- (c) PLRU weniger Misses verursacht als LRU
- (d) PLRU weniger Misses verursacht als FIFO.

Markieren Sie jeweils welche Zugriffe Misses verursachen.

Aufgabe 11.3: Betriebssysteme

Wir haben in der Vorlesung drei Aufgabengebiete eines Betriebssystems motiviert: Virtualisierung, Nebenläufigkeit, Persistenz. Erläutern Sie zunächst kurz diese Begriffe. Beschreiben Sie dann jeweils, welche konkreten Teilaufgaben sich dahinter verbergen und warum es nützlich ist, dass sich ein Betriebssystem darum kümmert.

Hinweis: Als Hintergrundliteratur empfehlen wir das frei verfügbare Buch Operating Systems: Three Easy Pieces unter http://pages.cs.wisc.edu/~remzi/OSTEP/. Wir werden in der Vorlesung aber lediglich einige Teile des Buchinhaltes abdecken.



System Architecture SS 2021

Assignment 11

You may submit your solutions via the CMS until **10:00 a.m. on Wednesday, July 7, 2021**. Please state on your solutions your tutorial, and the names and matriculation numbers of all team members.

Problem 11.1: Pseudo-LRU

To realize the LRU replacement policy for a k-way fully associative cache, at least $\log(k!)$ bits are necessary to store the order of the cache elements with respect to their last access. In practice, therefore, one often uses pseudo-LRU (PLRU), which behaves similarly to LRU but requires only k-1 bits.

To illustrate how the pseudo-LRU policy works, we assume that these bits are arranged in a tree-like fashion, as shown in Figure 2. This implies a path leading to the next cache element to be replaced. Here, a 1 (0) means that the right (left) branch is taken.

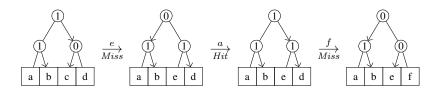


Figure 2: Behavior of a 4-way fully associative PLRU cache for the access sequence e, a, f.

After an access (hit or miss), the bits on the path from the root to the accessed element are adjusted so that they point away from the accessed element. Figure 2 shows a sequence of three such accesses.

Starting from the right PLRU cache state in Figure 2, execute the access sequence

adbcdefead

State the respective intermediate steps (cache states) explicitly.

Problem 11.2: Comparison of Replacement Policies

Given an empty, 4-way fully associative cache. Specify an access sequence for each case such that

- (a) LRU causes fewer misses than FIFO
- (b) FIFO causes fewer misses than LRU
- (c) PLRU causes fewer misses than LRU
- (d) PLRU causes fewer misses than FIFO.

Indicate for each case which accesses cause cache misses.

Problem 11.3: Operating Systems

We motivated three tasks of operating systems in the lecture: Virtualization, Concurrency, Persistence. First, briefly explain these terms. Then describe for each term which concrete subtasks are hidden behind them and why it is useful that an operating system takes care of them.

Note: As background literature, we recommend the freely available book Operating Systems: Three Easy Pieces at http://pages.cs.wisc.edu/~remzi/OSTEP/. However, we will only cover some parts of the book's content in this course.