



Bitte bearbeiten Sie die folgenden Aufgaben in Zweiergruppen. Wenn Sie Feedback zu Ihren Lösungen haben möchten, können Sie diese im Anschluss an die Übung auf Papier oder in elektronischer Form per Mail an h.g.esser@gmx.de abgeben (bitte keine Mehrfachabgabe identischer Lösungen). Es gibt keine Bewertung/Benotung.

1. Cache, LRU, Pseudo-LRU

a) Erläutern Sie (stichwortartig) den in der Vorlesung besprochenen Unterschied zwischen LRU und Pseudo-LRU.

b) Wie viele verschiedene Zustände können LRU und Pseudo-LRU (bei n Blöcken in einem Set) speichern? Für welche Werte n sind die Strategien LRU und Pseudo-LRU identisch?

c) Implementieren Sie LRU und Pseudo-LRU für ein einzelnes Set, das aus acht Blöcken besteht (wie Beispiel der Vorlesung), in Python (oder einer anderen Programmiersprache).

(i) Überlegen Sie sich dazu zunächst eine sinnvolle Datenstruktur für eine Sequenz von Zugriffen.

Zugriffe haben eine der beiden Formen

1. **READ** i (liest den i -ten Block)

2. **STORE** (verdrängt einen Block – welchen, muss Ihr Programm entscheiden)

(ii) Überlegen Sie, wie eine initiale Belegung der Zugriffsreihenfolge (LRU) bzw. des Entscheidungsbaums (Pseudo-LRU) aussehen kann. Die beiden Initialbelegungen sollen einander (so weit wie möglich) entsprechen.

(iii) Gestalten Sie das Programm so, dass für eine eingegebene Sequenz von Zugriffen (Liste der **READs** und **STOREs**) sowohl die LRU- als auch die Pseudo-LRU-Strategie verfolgt werden kann; die Strategie soll also ein Parameter sein.

(iv) Dokumentieren Sie Ihr Programm, testen Sie es mit verschiedenen Zugriffssequenzen und erstellen Sie ein Ergebnisprotokoll. Gehen Sie dabei auch auf Unterschiede zwischen LRU und Pseudo-LRU ein.