

## Portfolioprüfung – Werkstück A – Alternative 3

### Aufgabe 1    Aufgabe

Entwickeln und implementieren Sie einen Simulator für die Seiteneretzungsstrategien **Optimal**, **Random**, **LFU** (Least Frequently Used), **LRU** (Least Recently Used), **FIFO** (First In First Out) und **Clock/Second Chance**.

Der Simulator soll einen Speicher mit einer bestimmten Anzahl an Seiten darstellen und die Belegung bzw. Freigabe sowie Zugriffe auf Seiten korrekt simulieren und demonstrieren. Die Anzahl der Seiten können die Benutzer (mit sinnvollen Einschränkungen!) frei festlegen.

**Entwickeln und implementieren Sie Ihre Lösung als Bash-Skript oder als C-Programm** als freie Software (Open Source) und verwenden Sie hierfür ein Code-Repository, z.B. bei GitHub.

Bearbeiten Sie die Aufgabe in Teams zu **4 Personen**.

Schreiben Sie eine aussagekräftige und ansehnliche Dokumentation (Umfang: **8-10 Seiten**) über Ihre Lösung.

Bereiten Sie einen Vortrag mit Präsentationsfolien und eine Live-Demonstration (Umfang: **15-20 Minuten**) vor. Demonstrieren Sie die Funktionalität der Lösung in der Übung.

### Aufgabe 2    Anforderungen an den Simulator

- Das fertige Programm soll eine Kommandozeilenanwendung sein.
- Der Quellcode soll durch Kommentare verständlich sein.
- Benutzer sollen die Größe des gesamten Speichers über eine Benutzereingabe (interaktiv) oder per Kommandozeilenargument definieren, also z.B. `-size 10`.
- Benutzer sollen die Seiteneretzungsstrategien Optimal, Random, LFU, LRU, FIFO und Clock/Second Chance über eine Benutzereingabe (interaktiv) oder per Kommandozeilenargument definieren, also z.B. `-strategy FIFO`.
- Es soll auch möglich sein mehrere Seiteneretzungsstrategien gleichzeitig auszuwählen (als Benutzereingabe oder als Kommandozeilenargument) und so verschiedene Seiteneretzungsstrategien nacheinander zu simulieren.
- Benutzer sollen über eine Benutzereingabe (interaktiv) über Tastatureingaben nach Belieben den Zugriff auf Speicherseiten und damit auch das Nachladen und Ersetzen von Speicherseiten simulieren können.

- Es soll möglich sein, alternativ zur interaktiven Benutzereingabe, die Anforderungssequenz der Speicherseiten über eine einzulesende Eingabedatei zu definieren. Konkret sollen die Benutzer die Möglichkeit haben, per Kommandozeilenargument den Pfad und Dateinamen der Textdatei zu definieren, also z.B. `-sequencefile <dateiname>`.
- Nach jeder Änderung der Speicherbelegung soll der Simulator Informationen zur Effizienz der ausgewählten Seitenersetzungsstrategie(n) in der Kommandozeile ausgeben. Sinnvolle Informationen sind u.a.:
  - Hitrate
  - Missrate
- Die Ausgabe des Simulators soll in eine Textdatei ausgegeben werden, deren Pfad und Dateiname die Benutzer frei definieren, also z.B. `-ausgabedatei <dateiname>`.

## Aufgabe 3    Literatur

- Foliensatz 2 der Vorlesung **Betriebssysteme und Rechnernetze** im SS2022
- **Betriebssysteme kompakt**, *Christian Baun*, 2. Auflage, Springer Vieweg, S. 105-111
- **Betriebssysteme**, *Carsten Vogt*, 1. Auflage, Spektrum Akademischer Verlag (2001), S. 162-163
- **Betriebssysteme – Prinzipien und Umsetzung**, *William Stallings*, 4. Auflage, Pearson (2003), S. 413-420
- **Betriebssysteme**, *Eduard Glatz*, 4.Auflage, dpunkt (2019), S.479-486
- **Grundkurs Betriebssysteme**, *Peter Mandl*, 4.Auflage, Springer Vieweg (2014), S.236-2476
- **Moderne Betriebssysteme**, *Andrew Tanenbaum*, 3.Auflage, Pearson (2009), S.255-269