

# Verständliche Erklärung des Projekts

## **Konzept und Zielsetzung der Speicherverwaltungssimulation**

Dieses Projekt stellt eine Simulation zur Speicherverwaltung dar. Im Mittelpunkt steht der Vergleich verschiedener Seitenersetzungsalgorithmen, wobei die Anzahl der entstandenen Seitenfehler als Bewertungsmaßstab dient. Das Projekt bietet eine Schnittstelle, über die neue Seitenersetzungsalgorithmen jederzeit einfach implementiert und getestet werden können. Die Simulation basiert auf dem Konzept der Diskreten Ereignissimulation (DES).

## **Komponenten der Simulation**

### Algorithmen.h / .cpp

Hier werden alle Seitenersetzungsalgorithmen implementiert. Sie erben von der Oberklasse Pagingalgorithmus. Je nach Funktionsweise des jeweiligen Algorithmus werden die Methoden `on_page_access()`, `on_page_fault()` und `select_victim()` individuell überschrieben.

### Discrete\_event\_simulation.h (namespace des)

Enthält alle Bestandteile, die für die Diskrete Ereignissimulation benötigt werden:

- Event
- EventQueue

Wichtig ist, dass jedes Event einen Zeitstempel besitzt, sodass die Events in der EventQueue zeitlich sortiert verarbeitet werden können.

### Simulation.h / .cpp (namespace sim)

Beinhaltet ein Statistikmodul, das folgende Kennzahlen erfasst:

- Anzahl der Seitenzugriffe
- Anzahl der TLB-Hits
- Anzahl der Seitenfehler

Außerdem enthält dieser Teil den Kern der Simulation:

- TLB und TLBEntry
- Seitentabelle (Pagetable) und PagetableEntry
- MMU (Memory Management Unit)
- Prozess

## **Ablauf der Simulation**

Zu Beginn werden benutzerdefinierte Parameter gesetzt:

- Größe des TLBs
- Größe des physischen Speichers
- Größe des virtuellen Speichers

Anschließend werden die Events nacheinander aus der EventQueue entfernt und mithilfe der Methode `MMU::handleEvent()` verarbeitet. Der Ablauf wird dabei ausführlich auf der Konsole dokumentiert, sodass eine Nachverfolgung möglich ist. Nach der Verarbeitung aller Events wird die Statistik zu Seitenzugriffen, TLB-Hits und Seitenfehlern ausgegeben.

## **Zusätzliche Informationen**

In diesem Projekt wird die Übersetzung von virtuellen zu physischen Adressen nicht behandelt. Stattdessen erfolgt eine Abstraktion, bei der ausschließlich mit Seitennummern und Rahmennummern gearbeitet wird.

Ein Prozess greift also nicht auf eine konkrete Speicheradresse zu, die einer virtuellen Seite zugeordnet wird, sondern es wird direkt von den virtuellen Seiten ausgegangen, auf die zugegriffen wird.