

01. Übungsblatt : Betriebssysteme

Ausgabe: Dienstag, 21. Oktober 2025

Dr. rer. nat. Bernhard J. Berger

Bearbeitungszeitraum: 21.10. bis 04.11. 2025

Abgabe: 05. November 2025

Wintersemester 2025/2026

Auf diesem Übungsblatt werden wir uns zunächst noch mit Programmierung im User-Space auseinandersetzen. Konkret werden wir unterschiedliche Arten ausprobieren, um ein Problem parallel zu lösen.

Aufgabe 1: Schnelle Matrizenmultiplikation

In dem ZIP-Archiv `sequential.zip` befindet sich das Programm `sequential`. Das Programm erzeugt zunächst eine Matrix A mit den Dimensionen $\text{dimM} \times \text{dimN}$ vom Typ `double` mit `dimM` Zeilen und `dimN` Spalten. Anschließend wird eine Schleife `numMultiplications` mal ausgeführt. In jedem Schleifendurchlauf werden die folgenden Schritte ausgeführt:

1. Eine Matrix V der Größe $\text{dimN} \times \text{dimM}$ wird mit zufälligen `double` Werten gefüllt.
2. Das Matrixprodukt $C = A \cdot B$ wird berechnet.

Die Aufgabe: Parallelisierung mit POSIX-Threads. Schreiben Sie ein neues Programm `threads`, welches die gleiche Aufgabe wie das Programm `sequential` löst. Hierbei soll jedoch Parallelisierung eingesetzt werden und die Ausführungszeit verglichen werden. Beim Design des Programms `threads` beachten Sie bitte die folgenden Hinweise:

1. Die `main`-Funktion erzeugt die initiale Matrix A .
2. Die `main`-Funktion führt eine Schleife `numMultiplications` aus und führt in jedem Schleifendurchgang die folgenden Dinge aus:
 - (a) Die Matrix B wird mit zufälligen Werten gefüllt.
 - (b) Eine neue Implementierung der Funktion `multiplyAB()` wird aufgerufen. Diese Funktion aktiviert k Threads mit den logischen IDs 0 bis $(k - 1)$.
 - (c) Thread i , $i = 0, \dots, k - 1$ berechnet die Reihen $i + j \cdot k$, $j = 0, 1, \dots$ der Matrix $C = A \cdot B$.

Die Auswertung: Vergleich die Ausführungszeit von `sequential` mit der Ausführungszeit von `threads` für unterschiedliche Werte für k . Stellen Sie die Ausführungszeiten grafisch gegenüber und geben Sie eine Erklärung für das optimale k .

Hinweise:

- Um das Programm besser zu testen ist es sinnvoll die Matrizengröße zu verkleinern und die Zufallszahlen durch feste Werte auszutauschen.
- Eine Hilfsfunktion zum Ausgeben der Matrizen A , B und C vereinfacht die Fehlersuche.

Aufgabe 2: Schnelle Matrizenmultiplikation 2

Die Aufgabe: Parallelisierung mit User Threads. Implementieren Sie das Programm `user`, welches das Problem aus Aufgabe 1 mit User Threads löst.

Evaluation: Vergleich die Ausführungszeit aller Lösungen für unterschiedliche Werte für k und stellen Sie die Ausführungszeiten gegenüber. Visualisieren Sie auch hier die Ergebnisse.