

# Übungsblatt 03

André Reissig

Benedikt Peterson

3. Dezember 2025

## Aufgabe 1

**Grenzen Sie die folgenden Begriffe (gegenseitiger Ausschluss, Signalisierung, Synchronisation, Koordination, Kommunikation, Kooperation) auf Grund der Vorlesung gegeneinander ab? Gehen Sie bei Ihrer Diskussion insbesondere auf die Teilnehmer und ihre Beziehung zueinander ein.**

Die Aspekte bei der Zusammenarbeit von mehreren Threads lassen sich in zeitliche und funktionale unterteilen. Zu den funktionalen gehört die Kommunikation, das Schicken von Daten, und die Kooperation, der Zugang zu gemeinsamen Daten und zum zeitlichen Aspekt gehört die Koordination, wann arbeitet welcher Thread, und die Synchronisation, gleichschaltung der Threads um eine Koordination zu ermöglichen.

Da die Synchronisation ein Teil der Koordination ist, spricht man bei der Kommunikation, Kooperation und Koordination von den drei Grundformen, dabei ist die fundamentalste die Koordination, welche benötigt ist für Kooperation und Kommunikation (Ohne Zuordnungen, können keine Daten verschickt oder geteilt werden). Wenn verschiedene Threads miteinander arbeiten, nennt man das Interaktion. Die einfachste solcher Interaktionen ist die Signalisierung. Die Signalisierung wird benutzt, um eine zeitliche Ordnung zwischen Aktivitäten einzustellen.

Beispiel: Aktivität 1 wartet solange bis das Signal von Aktivität 2 kommt.

Wenn nun Aktivität 2 genauso eine wait Funktion implementiert hat vor ihrem sendenden Signal und damit auch auf Aktivität 1 warten muss haben wir einen gegenseitigen Ausschluss aufgebaut. In einer solchen Situation darf die Aktivität, die als erstes bei ihrer "wait" Funktion angekommen ist, passieren und blockiert erst dann die andere Aktivität. Dies ist wichtig, wenn es egal ist, welche Aktivität als erstes passiert, weil beide (z.B. gemeinsam) benötigt werden, aber beide nicht zusammen durchgeführt werden können. Der gegenseitige Ausschluss ist ein Teil um eine Koordination zwischen Threads herzustellen.

Eine Synchronisation erzeugt man, indem man Signalisierung symmetrisch zwischen Threads aufbaut. Durch Synchronisation zum Beispiel lassen sich auch eine andere Unterart der Kommunikation erstellen, die Synchron Kommunikation. Weitere Arten sind die Koordinierte und Asynchrone Kommunikation. Sie unterscheiden sich in der Art wie die send und "receive" Befehle benutzt werden (synchron, asynchron oder in einer festen Ordnung). Kommunikation muss nicht nur 1 zu 1 stattfinden, weiter funktioniert es auch bei 1-n, n-1 und n-m.

Kooperation ist wie oben schon erwähnt, wenn mehrerer Threads Zugriff auf gemeinsame Daten haben, um Fehler und Inkonsistenzen vorzubeugen. Um das zu gewährleisten ist Koordination essenziell. Ein wichtiger Bestandteil für Kooperation sind "locks", also wer darf auf was zugreifen, wann und auf was nicht. Charakterisierend für Kooperation ist auch der Fakt das zu jeder Zeit ein Thread etwas durchführt (schreibt). Dort lässt sich auch gut auf Reader/Writer Kooperation eingehen, denn wenn viele Threads auf gleiche Daten zugreifen können, jedoch nur ein Thread berechtigt ist zu schreiben, lässt sich dies einfacher Koordinieren. Kooperation lässt sich auch in 1-1, n-1, 1-n, n-m Beziehungen durchführen.

## Aufgabe 2

Die Programme sind zu finden unter dem folgenden link

[https://github.com/benedikt-code/bs\\_ws\\_25-26](https://github.com/benedikt-code/bs_ws_25-26)

Die Dateien sind zu finden in dem Ordner **Uebung3**. Die Programme sind im Unterordner **Core/src**.

Als verwendete Hilfsmittel ist ChatGPT zu nennen. ChatGPT wurde benutzt, um eine erste, grobe Idee zu bekommen, wie eine Implementierung aussehen könnte.