– Praktikumsaufgabe 9 –

Thema: *Parallele Programmierung*

Zielstellung: Erlernen der Grundelemente paralleler Programmierung in Rust: Erzeugung von Threads, Nutzung von `join()`, Kommunikation über *Channels*. Parallelisierung der Berechnung des Apfelmännchens.

1. Finden Sie experimentell heraus, wie viele Threads Sie mittels `spawn()` auf ihrem System erzeugen können. Gibt es Auswirkungen auf das Systemverhalten?

2. Entwickeln Sie ein Rust-Programm, das drei Threads startet. Der erste Thread soll die Zahlen von 1 bis zu einem konfigurierbaren Maximum (z. B. 100) generieren und diese über einen *Channel* an den zweiten übermitteln. Der zweite Thread soll jeweils zwei empfangene Zahlen miteinander addieren und die Summe über einen zweiten *Channel* an den dritten Thread übermitteln. Der dritte Thread soll jeweils zwei aufeinanderfolgende empfangene Summen miteinander multiplizieren und die Produkte jeweils auf den Bildschirm schreiben.

   Hinweis: Beachten Sie, dass die `recv()`-Operation ein `Result<T, RecvError>` zurückliefert. Falls im sendenden Thread das entsprechende Handle vernichtet wird, weil der Thread das Senden beendet hat, ist das Resultat `Err(RecvError)`.

3.* Versuchen Sie, die Berechnung des Apfelmännchens aus Aufgabenblatt 2+ zu beschleunigen, indem Sie mehrere Threads nutzen. Gelingt es Ihnen, die Anzahl der Threads konfigurierbar zu halten?

   Hinweise:

   - Jeder Thread könnte beispielsweise eine gewisse Anzahl Zeilen berechnen und das Ergebnis in einem „Teilbild“ ablegen.
   - Nach Beendigung der Berechnung sendet jeder Thread sein „Teilbild“ an den `main()`-Thread, der die empfangenen Teile in den Framebuffer kopiert und anschließend mittels `window.update_with_buffer()` dessen Ausgabe vornimmt. Dieses Modell (mehrere Sender, ein Empfänger) lässt sich prima auf das `std::sync::mpsc`-Crate abbilden.
   - Eine Schwierigkeit besteht darin, dass die Laufzeit der Threads differiert, und somit der Empfang der „Teilbilder“ nicht unbedingt in der korrekten Reihenfolge garantiert ist.
   - Gehen Sie inkrementell vor; zunächst sind zwei Threads völlig ausreichend.