Engineer Software

Swift-6-Nebenläufigkeit über C abgesichert

Vollständige strikte Nebenläufigkeit von Swift 6 ohne eigene Actors eingeführt und eine blockierende C-Ereignisschleife über genau einen Produzenten, einen Verbraucher und eine Reihenfolge an den Main Actor angebunden, nachdem ein Task je Ereignis die für die Oberfläche nötige Reihenfolge verloren hatte.

September 2026

strikte Nebenläufigkeit mit einer Reihenfolge und ohne eigene Actors

Situation. Die vollständige strikte Nebenläufigkeitsprüfung von Swift 6 macht aus Datenrennen Compilerfehler statt gelegentlicher Abstürze. Bei einer C-Bibliothek wird ihre Einführung schwierig: Die Kernereignisschleife ist ein dauerhaft außerhalb des Hauptthreads laufender blockierender Aufruf. Ihre Rückgabewerte sind Pointer ohne Nebenläufigkeitsgarantien. Der Compiler kann darüber nichts ableiten und verweigert die Nutzung bis zur ausdrücklichen Beschreibung der Grenze.

Aufgabe. Die vollständige Prüfung musste ohne Ausweichmöglichkeiten aktiviert werden. Dafür brauchte es einen entworfenen Übergang von der blockierenden C-Schleife zum Main Actor statt umgehender Annotationen.

Massnahmen. Der naheliegende Ansatz eines Actors je Teilsystem wurde nach Messung statt Geschmack verworfen. Ein Task je Ereignis lässt die Laufzeit beliebig umordnen, während der Kern geordnete Ereignisse liefert. Überholt eine Nachrichtenänderung ihre Nachrichtenerstellung, zeigt die Oberfläche eine Änderung an einer noch nicht existierenden Nachricht. Actor-Reentranz verschärft das: Ein Actor kann mitten in einer Methode pausieren und einen anderen Aufruf verarbeiten. Die Alternative ist bewusst schlicht: ein Produzententhread für die blockierende Schleife, ein Verbraucher, eine Queue dazwischen und am Ende genau ein Übergang zum Main Actor. Die Reihenfolge bleibt erhalten, weil nur ein Pfad existiert und nichts überholt.

Unsichere grenzüberschreitende Typen sind in Hülltypen gekapselt. Deren Threadsicherheit wird an der Hülle ausdrücklich zugesichert und begründet statt vorausgesetzt: Ein Thread besitzt den Pointer, vor der Übergabe wird kopiert.

Ergebnis. Die Anwendung kompiliert unter vollständiger strikter Nebenläufigkeitsprüfung ohne Unterdrückungen. Ereignisreihenfolge ist strukturell gesichert statt erhofft. Der Preis ist weniger Parallelität als möglich: Alles läuft durch einen Verbraucher. Wird er zum Engpass, muss erneut hergeleitet werden, welche Ereignisse sich sicher umordnen lassen — genau die hier vermiedene Analyse.

Diese Erfahrung für Ihr Projekt nutzen

Nutzen Sie „Swift-6-Nebenläufigkeit über C abgesichert“ als Ausgangspunkt für eine ähnliche Herausforderung in Ihrer Organisation. Die Leistungen hinter dieser Arbeit stehen auch für Ihr Projekt zur Verfügung. Die Struktur eines Systems wird zuerst auf dem Papier durchdacht, damit die Teile zusammenpassen und die Kosten feststehen, bevor die erste Zeile Code entsteht.

Unsere Leistungen entdecken Kontakt aufnehmen →

Hinter das Ergebnis schauen

Lesen Sie die Situation, die Entscheidungen und das Ergebnis dieses Beispiels. Wir können besprechen, welche Teile sich auf Ihre Rahmenbedingungen übertragen lassen und welche einen anderen Ansatz erfordern. Swift-6-Nebenläufigkeit über C abgesichert — strikte Nebenläufigkeit mit einer Reihenfolge und ohne eigene Actors

Unsere Leistungen entdecken Kontakt aufnehmen →

Möchten Sie Nebenläufigkeit, deren Sicherheit der Compiler prüft?

Besprechen Sie Ihr Projekt mit der Ingenieurin, die für die Umsetzung verantwortlich ist. Wir vereinbaren Umfang, Preisgrundlage und Übergabe schriftlich, bevor die Arbeit beginnt.

Rollenkontext: macOS-Anwendungsingenieurin. Die Arbeit baut auf Lösungsarchitektur, Backend-Entwicklung, Frontend-Entwicklung, Zuverlässigkeit und Sicherungen, Performanceoptimierung und Tests und Qualitätssicherung. Und wenn Sie schon hier sind: Wir sind auch ziemlich gut in Plattform- und Lösungsarchitektur, Backend- und API-Entwicklung und Technische Leitung und Beratung.

Kontakt aufnehmen Unsere Leistungen entdecken

Ein Teil einer **längeren Arbeitsbilanz** — vollständig auf dieser Website zu finden.

Jeder Eintrag folgt demselben Aufbau: die Situation, die Aufgabe, unser Vorgehen und die Veränderungen.

Alle Arbeiten ansehen

Die benötigte Verbesserung definieren

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Möchten Sie Nebenläufigkeit, deren Sicherheit der Compiler prüft?

Fragen Sie nach den Entscheidungen hinter „Swift-6-Nebenläufigkeit über C abgesichert“. Sie sprechen mit den Menschen, die die Arbeit erklären und Ihr eigenes Projekt einschätzen können.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://software.engineer.company/de/portfolio/adopted-swift-6-strict-concurrency-over-c/