Engineer

C上にSwift 6の厳密並行処理を採用

インターフェースに必要な順序がイベントごとのタスク生成で失われると確認し、独自アクターを用いずSwift 6の完全な厳密並行処理を採用。ブロッキングCイベントループとメインアクターを、生産者1つ、消費者1つ、順序1つで接続した。

9月 2026

独自アクターを使わず順序を1つにした厳密並行処理

状況. Swift 6の完全な厳密並行処理検査は、データ競合を断続的なクラッシュでなくコンパイルエラーにする。ただしCライブラリ相手では難しく、ブロッキングするコアのイベントループを主スレッド外で永続実行し、返る値は並行性保証のないポインターである。境界を明示しなければ、コンパイラーは推論できず、すべてを拒否する。

課題. 注釈で回避せず、ブロッキングCループからメインアクターへの受け渡しを設計し、抜け道なしで完全検査を有効にする必要があった。

行動. 部分ごとのアクターという明白な方法を、好みでなく測定に基づき退けた。イベントごとにタスクを生むと任意順序に実行できるが、コアは順序付きであり、変更が作成を追い越せば、まだないメッセージの編集を表示する。アクターもメソッド途中で中断して他を処理する再入可能性により悪化させる。代わりに、ループを所有する1生産者スレッド、1消費者、その間の1キュー、最後のメインアクターへの1回の移動という単純な形にした。経路が1つで追い越せないため順序が残る。危険な値はラッパーの型で包み、安全を想定せず、1スレッドが所有し渡す前にコピーするという理由を文書化して、ラッパーでスレッド安全性を表明した。

結果. 抑制なしの完全検査でコンパイルでき、順序は期待でなく構造となった。一方、全体が1消費者を通るため可能なほど並列ではない。ここが制約になれば、安全に並べ替えられるイベントを導出し直す必要がある。それは、まさに今回避けた分析だった。

この経験をお客様のプロジェクトに

「C上にSwift 6の厳密並行処理を採用」を、お客様の組織の似た課題を考える出発点にしてください。この実績を支えたサービスを、お客様のプロジェクトにも提供できます。 まず全体像を設計し、各部分のつながりと費用を、コードを書き始める前に明らかにします。

サービスを見る お問い合わせ →

結果の背景を確かめる

この事例の状況、判断、結果をご覧ください。どの部分がお客様の制約の下でも生かせるか、どこに別の方法が必要かをご相談いただけます。 C上にSwift 6の厳密並行処理を採用 — 独自アクターを使わず順序を1つにした厳密並行処理

サービスを見る お問い合わせ →

コンパイラーが安全性を検証できる並行処理を求めていますか？

納品を担当するエンジニアに、プロジェクトについてご相談ください。着手前に、業務範囲、価格の算定方法、引き継ぎ内容を書面で合意します。

担当した役割：macOSアプリケーションエンジニア。 関連する分野： ソリューション設計, バックエンド開発, フロントエンド開発, 信頼性・バックアップ, 性能改善 および テスト・品質保証. こちらのサービスも得意としています プラットフォーム・ソリューション設計, バックエンド・API開発 および 技術面の統率・コンサルティング.

お問い合わせ サービスを見る

より多くの実績の **一つです** — そのすべてをこのサイトで公開しています。

どの項目も、状況、課題、実施内容、変化という同じ構成で記載しています。

すべての実績を見る

必要な改善を定める

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

コンパイラーが安全性を検証できる並行処理を求めていますか？

「C上にSwift 6の厳密並行処理を採用」の判断の背景についてお尋ねください。仕事内容を説明し、お客様のプロジェクトを検討できる担当者が対応します。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

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

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/portfolio/adopted-swift-6-strict-concurrency-over-c/