**Engineer** Консультування

# Увімкнула сувору конкурентність Swift 6 поверх C

Перейшла на повну сувору перевірку конкурентності Swift 6 без акторів, з'єднавши блокувальний цикл подій C з головним актором через одного виробника, одного споживача й один порядок, — з'ясувавши, що окрема задача на кожну подію губить порядок, від якого залежить інтерфейс.

вересень 2026

## сувора конкурентність з одним порядком і без акторів

**Ситуація.** Повна сувора перевірка паралельності у Swift 6 перетворює гонки даних на помилки компіляції замість епізодичних аварійних завершень. Складнощі починаються, коли її застосовують до бібліотеки на C: цикл подій ядра — це блокувальний виклик, який має безперервно виконуватися поза головним потоком, а значення, які він повертає, — вказівники без жодних гарантій щодо паралельності. Компілятор не може зробити щодо цього жодних висновків і відхилятиме все, доки межу не буде описано явно.

**Завдання.** Повну перевірку треба було ввімкнути без жодних лазівок, а отже — спроєктувати перехід від блокувального циклу на C до головного актора, а не обходити проблему анотаціями.

**Дія.** Очевидний підхід — окремий актор для кожної підсистеми, але його відкинули на підставі вимірювань, а не з міркувань смаку. Якщо створювати окрему задачу для кожної вхідної події, середовище виконання може планувати їх у довільному порядку, а потік подій ядра впорядкований: якщо подія «повідомлення змінено» випередить подію «повідомлення створено», на яку посилається, інтерфейс покаже редагування того, чого ще не існує. Повторний вхід (reentrancy) в актора лише погіршує ситуацію, бо актор може призупинитися посеред методу й обробити інший виклик. Замість цього обрали навмисно просту модель: один потік-виробник, який володіє блокувальним циклом, один споживач,

одна черга між ними й один-єдиний перехід на головного актора наприкінці. Порядок зберігається, бо шлях рівно один і ніщо нічого не випереджає. Небезпечні типи, що перетинають цю межу, загорнуто в типи, потокобезпеку яких явно гарантує обгортка, а не просто припускають, і це твердження задокументовано разом із поясненням, чому воно справджується: вказівником володіє один потік, і його копіюють, перш ніж передати.

**Результат.** Застосунок компілюється з повною суворою перевіркою конкурентності без жодних придушень, а порядок подій гарантує сама структура, а не надія. Ціна в тому, що архітектура менш паралельна, ніж могла б бути: усе проходить через одного споживача, і якщо він колись стане вузьким місцем, для виправлення доведеться заново з’ясовувати, які події можна безпечно переставляти, — а це саме той аналіз, якого вдалося уникнути.

Застосуйте цей досвід у своєму проєкті

Використайте «Увімкнула сувору конкурентність Swift 6 поверх C» як відправну точку для схожого завдання у вашій організації. Послуги, що стоять за цією роботою, доступні й для вашого проєкту. Хтось на вашому боці столу, коли підрядник, бюджет або архітектурне рішення потребують кваліфікованої оцінки.

**Переглянути наші послуги** Зв'язатися з нами →

Дізнайтеся, що привело до результату

Прочитайте про ситуацію, рішення та результат цього прикладу. Ми можемо обговорити, що відповідає вашим умовам, а що потребує іншого підходу. Увімкнула сувору конкурентність Swift 6 поверх C — сувора конкурентність з одним порядком і без акторів

**Переглянути наші послуги** Зв'язатися з нами →

## Хочете конкурентність, безпеку якої доведе компілятор?

Обговоріть свій проєкт з інженером, відповідальним за виконання. Обсяг, основу розрахунку вартості та передачу результатів узгоджуємо письмово до початку робіт.

Роль у цій роботі: Інженер застосунків macOS. Тут задіяні Архітектура рішень, Backend-розробка, Frontend-розробка, Надійність і резервне копіювання, Оптимізація продуктивності та Тестування та QA. І, до речі, ми добре вміємо Архітектура платформи та рішень, Backend- та API-розробка та Технічне лідерство та консалтинг.

**Зв'язатися з нами** Ознайомтеся з нашими послугами

## Одна частина **довшого переліку** — і він увесь на цьому сайті.

Кожен запис написано однаково: ситуація, завдання, що ми зробили і що змінилося.

**Переглянути весь перелік**

### Визначте потрібне покращення

Опишіть рішення, людей, яких воно стосується, та вже розглянуті варіанти. Ми можемо явно показати компроміси й залишити документовану рекомендацію.

**Дивитися, скільки це коштує** Зв'язатися з нами →

# Хочете конкурентність, безпеку якої доведе компілятор?

Запитайте про рішення в роботі «Увімкнула сувору конкурентність Swift 6 поверх C». Ви говоритимете з людьми, які можуть пояснити роботу й оцінити ваш проєкт.

**Переглянути наші послуги**

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

**Дбайливо зроблено з любов'ю — D.**

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

**Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.**

**Надруковано з https://advisory.engineer.company/uk/portfolio/adopted-swift-6-strict-concurrency-over-c/**