Engineer Rådgivning

Tok i bruk streng samtidighet i Swift 6 over C

Tok i bruk fullstendig streng samtidighet i Swift 6 uten actors, og koblet en blokkerende C-hendelsesløkke til main actor gjennom én produsent, én konsument og én rekkefølge — etter å ha slått fast at én task per hendelse mister rekkefølgen grensesnittet er avhengig av.

september 2026

streng samtidighet med én rekkefølge, og ingen actors

Situasjon. Den fullstendige, strenge samtidighetskontrollen i Swift 6 gjør datakappløp til kompileringsfeil i stedet for sporadiske krasj. Det er når den tas i bruk mot et C-bibliotek at det blir vanskelig: kjernens hendelsesløkke er et blokkerende kall som må kjøre utenfor hovedtråden for alltid, og verdiene den leverer tilbake, er pekere uten noen samtidighetsgarantier i det hele tatt. Kompilatoren kan ikke resonnere om noe av dette og avviser alt til grensen er beskrevet eksplisitt.

Oppgave. Fullstendig kontroll måtte slås på uten nødutganger, og det betydde å designe overgangen fra en blokkerende C-løkke til main actor i stedet for å annotere seg rundt den.

Handling. Den opplagte tilnærmingen er én actor per delsystem, og den ble forkastet på grunnlag av målinger, ikke smak. Å starte én oppgave per innkommende hendelse lar kjøremiljøet planlegge dem i vilkårlig rekkefølge, og kjernens hendelsesstrøm er ordnet — en hendelse om endret melding som kommer frem før hendelsen om opprettet melding som den viser til, gir et grensesnitt som viser en redigering av noe som ikke finnes ennå. Reentrancy i actorer gjør dette verre, ikke bedre, fordi en actor kan suspenderes midt i en metode og behandle et annet kall. Det som erstattet den, er bevisst enkelt: én produsenttråd som eier den blokkerende løkken, én konsument, én kø mellom dem og ett enkelt hopp over til main actor til slutt. Rekkefølgen bevares fordi det finnes nøyaktig én vei, og ingenting går forbi noe annet. De usikre typene som

krysser den grensen, er pakket inn i typer der trådsikkerheten erklæres i wrapperen i stedet for å antas, og erklæringen er dokumentert med hvorfor den holder — pekeren eies av én tråd og kopieres før den overleveres.

Resultat. Appen kompilerer med fullstendig streng samtidighetskontroll uten undertrykkinger, og hendelsesrekkefølgen er en strukturell egenskap og ikke et håp. Prisen er at designet er mindre parallelt enn det kunne vært: alt går gjennom én konsument, og hvis den konsumenten noen gang blir en flaskehals, vil løsningen kreve at man på nytt utleder hvilke hendelser som trygt kan stokkes om, og det er nøyaktig den analysen denne utformingen unngikk.

Bruk erfaringen i prosjektet deres

Bruk «Tok i bruk streng samtidighet i Swift 6 over C» som utgangspunkt for en lignende utfordring i organisasjonen deres. Tjenestene bak arbeidet er tilgjengelige for prosjektet deres. Noen på din side av bordet når en leverandør, et budsjett eller en arkitekturbeslutning trenger en ny, kyndig vurdering.

Se tjenestene våre Ta kontakt →

Se bak resultatet

Les om situasjonen, beslutningene og resultatet i eksemplet. Vi kan drøfte hva som passer deres rammer, og hva som krever en annen tilnærming. Tok i bruk streng samtidighet i Swift 6 over C — streng samtidighet med én rekkefølge, og ingen actors

Se tjenestene våre Ta kontakt →

Vil du ha samtidighet som kompilatoren kan bevise er trygg?

Snakk om prosjektet med ingeniøren som har ansvaret for leveransen. Vi avtaler omfang, prisgrunnlag og overlevering skriftlig før arbeidet begynner.

Rolle i oppdraget: Applikasjonsingeniør for macOS. Det bygger på Løsningsarkitektur, Backend-utvikling, Frontend-utvikling, Pålitelighet og sikkerhetskopier, Ytelsesoptimalisering og Testing og kvalitetssikring. Og nå som dere er her: vi er ganske gode på Plattform- og løsningsarkitektur, Backend- og API-utvikling og Teknisk ledelse og rådgivning.

Ta kontakt Se tjenestene våre

Ett stykke av **en lengre oversikt** — og den ligger i sin helhet her.

Hver oppføring er skrevet på samme måte: situasjonen, oppgaven, hva vi gjorde, og hva som endret seg.

Se hele oversikten

Definer forbedringen dere trenger

Ta med beslutningen, menneskene som berøres og alternativene som er vurdert. Vi kan tydeliggjøre avveiningene og etterlate en dokumentert anbefaling.

Se hva det koster Ta kontakt →

Vil du ha samtidighet som kompilatoren kan bevise er trygg?

Spør om beslutningene bak «Tok i bruk streng samtidighet i Swift 6 over C». Dere snakker med folk som kan forklare arbeidet og vurdere deres eget prosjekt.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://advisory.engineer.company/nb/portfolio/adopted-swift-6-strict-concurrency-over-c/