Engineer プラットフォーム

全FFI呼び出しを検証する構文解析器を作成

手書きの仮Cヘッダーで、削除済みの3関数を呼んでもコンパイル・リンクが通り実行時に落ちた問題を受け、実際の7,308行のCヘッダーを読み、全呼び出し箇所、列挙定数、ポインター所有クラスのfinal指定を検証する構文解析器を作成した。

9月 2026

7,308

推測せず読み取ったCヘッダーの行数

状況. 初期にはアプリが期待する関数を手書きのCヘッダーで表していた。コンパイルとリンクは成功したが、コアから消えた3関数を実際に呼ぶと落ちた。コンパイラーもリンカーも実ライブラリでなく説明に満足し、差は実行時だけに現れた。

課題. 7,308行・268宣言の実ヘッダーを正本とし、Swift内の全利用をレビューでなく自動照合する必要があった。

行動. 実際の上流ヘッダーを解析し、関数、列挙定数、型を集め、Swiftの全呼び出しと定数参照を照合した。未宣言関数や改名済み定数でビルドを失敗させる。268宣言中132を使用し、136が未使用と把握できること自体も、未到達のコア範囲を正確に示す。さらに、コアポインターを所有する全Swiftクラスをfinalにする、コンパイラーが強制しない規則を設けた。継承先が破棄処理や寿命を変えると、近くにunsafe語がなくてもuse-after-freeになるため、ツリー全体を名前で検査した。

結果. 古い参照は実行時でなくビルド時に失敗するため、元のクラッシュの種類を防げる。限界は宣言を理解しても意味は理解しないことであり、同名・同シグネチャのまま上流が意味や所有規則を変えても、無指摘で通る。

この経験をお客様のプロジェクトに

「全FFI呼び出しを検証する構文解析器を作成」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。 変更を検証し、リリースし、元に戻せる仕組みを整え、金曜の深夜ではなく火曜の午後にお客様へ届けられるようにします。

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。 全FFI呼び出しを検証する構文解析器を作成 — 7,308 推測せず読み取ったCヘッダーの行数

サービスを見る お問い合わせ →

削除された関数の呼び出しをビルドで検出できますか？

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： バックエンド開発, API・システム連携, 自動化・CI/CD, セキュリティ, テスト・品質保証 および 文書作成. こちらのサービスも得意としています バックエンド・API開発, DevOps・CI/CD自動化 および 技術文書の作成.

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

デプロイの流れ、運用上のリスク、緊急対応の担当者をお知らせください。検査、復旧手段、明確な責任の所在を伴う変更の範囲を定められます。

料金を見る お問い合わせ →

削除された関数の呼び出し をビルドで検出できますか？

「全FFI呼び出しを検証する構文解析器を作成」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://platform.engineer.company/ja/portfolio/wrote-a-parser-verifying-every-ffi-call/