



コンパイラ認定には 関数呼び出しのロバストなテストが不可欠



概要

言語仕様に対するコンパイラのテストは比較的分かりやすいと言えますが、ターゲットアーキテクチャの複雑な機能が正しく実行されることの保証は、はるかに困難です。言語規格は「関数呼び出し」がどれほど複雑になるかについて明確ではありませんので、「関数呼び出し」は、問題を引き起こすことが多い1つの領域です。複雑さはエラーを招くことになるため、そのテストには特に注意を払う必要があります。厳密にテストされていることが分かっている場合にのみ、真にコンパイラを信用できるのです。

関数の重要性はよく知られていることであり、アプリケーションをモジュール化して保守可能にする基本的な構成要素です。ある関数は、別の関数を呼び出してそれにデータを渡し、データを返してもらうことができます。このデータを格納する場所と方法はターゲットアーキテクチャに依存し、呼び出し規約によって定義されます。呼び出し規約は、ABI (Architecture Binary Interface) の重要な部分であり、プリミティブなデータ型と構造化データ型の表現も定義しています。関数呼び出しは非常に頻繁に発生することが常なので、呼び出し規約は速度を向上させるよう最適化されており、やはり複雑であるコンパイラのレジスタ割り当てと高度に結びついています。このような複雑さがあるにも関わらず、呼び出し規約は信頼できるインターフェースを提供しなければなりません。コンパイル済みのライブラリ関数や異なるコンパイラオプションでコンパイルされた関数が1つのアプリケーションにリンクされた場合にうまく結合して動作しなければなりません。

コンパイラ内、およびコンパイラ間のテスト

関数呼び出しをテストするには、呼び出し元と呼び出し先2つの関数が最低限必要です。この2つの関数を個別にコンパイルできる場合に呼び出し規約をテストする一つの方法は、それぞれの関数で異なるコンパイラオプションを使用し、2つの関数間で渡すパラメータの整合性を確認することです。アプリケーションのコードが現在使用されているコンパイラの古いバージョンでコンパイルされたライブラリとリンクされている場合、2つのテスト関数を同じコンパイラの2つのバージョンでコンパイルすることで呼び出し規約の安定性をテストできます。標準的なターゲットアーキテクチャでは、コンパイル済みのライブラリ関数が提供されることも多く、同じ呼び出し規約に従っていて相互に動作することを確認することは非常に重要です。



SuperTest Vermeer リリースのアップデート#1

SuperTest には既に、上記のような関数のペアを作成するジェネレーターがありましたが、その設定と使用はそれほど簡単ではありませんでした。SuperTest の新しいアップデート (SuperTest Vermeer リリースの最初のアップデート) において、これが劇的に変わりました。呼び出し規約テストは非常に使いやすくなり、コンパイラの検証中に使わない手はありません。これは、引数の値の範囲と戻り値の型が設定可能な関数ペアの広範な集合を生成し、限界まで呼び出し規約をテストします。

コンパイラ認定

呼び出し規約や ABI の他の部分は、C および C++ の言語規格では明示的に記述されていませんが、関数呼び出しは最も重要なプログラミング構成要素の一つです。関数呼び出しをその多様性すべてにおいて検証することは、コンパイラの信頼性を高める上で不可欠な要素であることに間違いはないでしょう。

顧客の声「ABI の規則に準拠することで、他のサプライヤーの製品と繋がることで問題が発生しても、お相手の問題として切り分けすることができている。各社が SuperTest を使ってくれれば、もっと助かるのだが、」

関数呼び出しのテストについて詳しくお知りになりたい場合や、コンパイラのテスト、SuperTest Vermeer リリースのアップデートについて他にご質問がありましたら、お問い合わせください。



富士設備工業株式会社 電子機器事業部 www.fuji-setsu.co.jp

(c) Copyright 2018 by Solid Sands B.V., Amsterdam, the Netherlands
SuperTest™ is a trademark of Solid Sands B.V., Amsterdam, The Netherlands.