

モダン・コンパイラ LLVM の基礎知識

FPR

モダン・コンパイラ LLVM とは

● はじまり

LLVM⁽¹⁾は、もともと Low Level Virtual Machine の略で、最初はイリノイ大学の学生による修士論文⁽²⁾として始まったプロジェクトです。それを元に、今(2020年)ではさまざまな言語およびアーキテクチャで再利用が可能なモジュール化されたコンパイラ、ツール・チェーン(デバッグなど)までを含む大きなプロジェクトに成長し、LLVMがプロジェクトを指す固有名詞となりました。

このプロジェクトは、コンパイラの開発を大きく便利にするためのインフラストラクチャとなるものであり、大学のさまざまな研究で使われる主なツールの1つとなっています。

● LLVMが変えるプログラミングの世界

LLVMの最大の利点は、コンパイラの開発コストを大幅に削減できることです。そのために、コンパイラや付随するツール・チェーンを言語非依存にモジュール化しており、これに重要な技術の1つとして、LLVM中間表現(LLVM IR)があります。

中間言語を用いないと、コンパイラの開発に多大な労力が掛かってしまいます(図1の左側)。それは、各高級言語に対して、各機械語へコンパイルするための実装をするため、実装にかかる労力が、

「高級言語の数」×「機械語の数」

となるからです。さらに、コンパイルするだけでなく、これら全てについてそれぞれの最適化アルゴリズムを実装しなければなりません。

一方で、中間言語を間に挟むと、各高級言語について、その中間言語へのコンパイル、そして中間言語から各機械語へのコンパイルのみを実装すればよくなります(図1の右側)。このため、実装にかかる労力は、

「高級言語の数」+「機械語の数」

となります。中間言語を用いない場合の掛け算よりはるかに小さい労力です。また、最適化のアルゴリズム

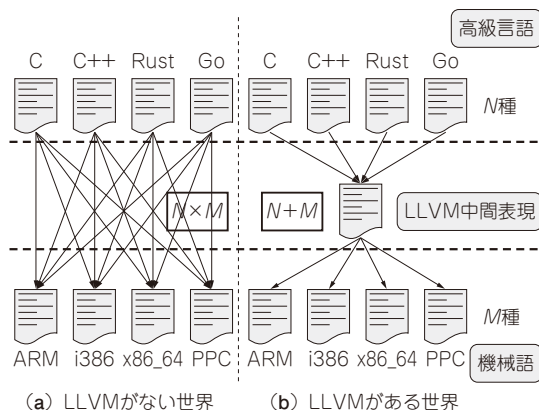


図1 モダン・コンパイラ LLVM はプログラミングの世界を大きく変える

も全て中間言語で実装するため、再利用が可能となり、機能のモジュール化が比較的容易になります。また、機械語依存の最適化を実装する場合は、高級言語とは独立に実装が可能です。

従って、LLVMには、コンパイラの研究および開発が大幅に楽になるという大きな利点があります。そのような事情により、近年では個人レベルでもコンパイラを開発できるという時代になりました。

例えば、新しい言語を作ろうと思ったとき、実装しなければならないのは、新しい言語から LLVM IR へのコンパイラのみです。最適化などは LLVM IR 上で行われるため、改めて実装する必要がありません。

さらに、企業が独自の命令セットを持つプロセッサを新たに作った場合、LLVM IR からそのプロセッサの機械語へのコンパイラを実装すれば十分です。

● ライセンス

LLVMはイリノイ大学のNCSAオープン・ソース・ライセンスに基づくものです⁽³⁾。つまり、LLVMを商用利用でき、企業にとってとても都合の良いものとなっています。

具体的には、LLVMを用いて作ったソフトウェア