

# JavaCC を用いた教育実習用の C b ミニコンパイラと仮想マシンインタプリタの試作

An implementation of C b language Mini-compiler and Virtual Machine Interpreter

CS 34 信田 裕介  
指導教員 吉村 晋

## 1. はじめに

コンパイラは, Java, C などのプログラム言語で書かれたソースコードを機械語のオブジェクトコードに変換するソフトウェアである. プログラムを作成・実行する際には, コンパイラを利用するが, その構造を理解していない. そこで実際にコンパイラの構造を理解するため, 具体的に試作する. ところでコンパイラ・コンパイラ (JavaCC) を用いた C b 言語 (C の縮小版) のミニコンパイラと, 仮想マシンインタプリタが作成され実験で使用されている [1, 2].

このコンパイラを改良するとともに, スタックマシンベースの仮想マシンを試作する.

## 2. JavaCC

コンパイラはソースコードを字句解析, 構文解析, 意味解析, コード生成の流れで処理する. JavaCC はコンパイラの開発を容易にするため字句解析, 構文解析部の構築を支援するツール (プリプロセッサ) で, Java 環境を前提とする [1]. 字句解析部, 構文解析部は JavaCC で前処理されて java ソースコードになり, コード生成部と同時にコンパイルされ, java 環境で実行できるコンパイラとなる.

## 3. C b コンパイラ

C b は C 言語のサブセット (縮小版) である [2]. C b コンパイラは, コンパイラ・インタプリタ方式を採用しており, ソースコードを VM (仮想マシン) コード (中間言語コード) にコンパイルし, VM インタプリタ上で実行することになる. ところで奥村 [2] は, 五月女 [1] のミニコンパイラと仮想マシンインタプリタを利用して C b 言語ミニコンパイラを作成した. しかし中間言語インタプリタは完全なスタックマシン対応でなかった. そこで疋田 [3] の提案した仮想マシンコードを採用し, コンパイラと仮想マシンインタプリタを試作する.

C b コンパイラとインタプリタは Fig. 1 の流れで実行される. ソースコードをコンパイラが受け取ると, 字句解析, 構文解析を行い, 逐次 VM コード (中間言語) を生成する. ソースを全て解析し終えたら VM コードを出力. 次に VM コードを仮想マシンに出力し, 受け取った VM コードを逐次実行する.

## 4. 仮想マシンインタプリタ

インタプリタはコンパイルで出力された VM コードを逐次解釈・実行するスタックマシン形式を採用する.

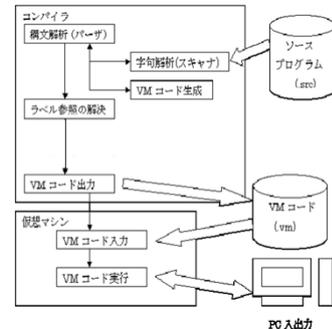


Fig. 1 コンパイラの流れ

スタックとは FILO の構造で保持されるデータ構造で, データをトップに追加する push と, トップからデータを引き出す pop の 2 つの基本操作を持つ.

Fig. 2 は形式を表す.

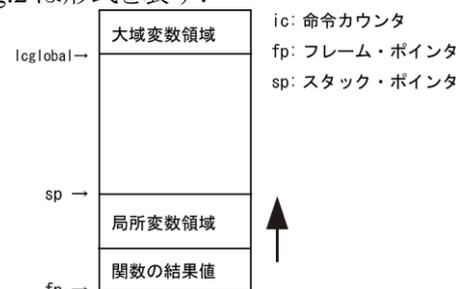


Fig. 2 スタックフレームの形式

## 5. 結果と考察

現在, コンパイラと仮想マシンの基本部分を作成し, デバック調整中である. 仮想マシンのスタックマシン部分は完成したが, 入出力対応部分, 関数呼び出し部分が未調整である.

当初の理解不足のため, インタプリタ試作に取り組むのが遅れたのが原因である. より早く計画的に考え, 実行していれば完成度を高められたと反省している.

## 6. おわりに

スタックマシンベースの仮想マシンの基本的な部分の試作が出来た. 今後は, 早急に未完部分を完成させるとともに, C b 言語の拡張機能の検討と実装する予定である.

## 文献

- [1] 五月女健治, “JavaCC コンパイラ・コンパイラ”, テクノプレス, 2003
- [2] 奥村真樹, 吉村晋, “JavaCC を用いたミニコンパイラの作成”, 都立航空高専 2008 年度卒業研究
- [3] 疋田輝夫・石畑清, “コンパイラの理論と実現”, 共立出版, 1988