

PASCAL処理系のFACOM230-38移植の経験と問題点

東京大学 工学部

武市 正人

富士通株式会社

高橋 博秋

1. はじめに

PASCAL処理系は、1971年言語の設計者N. Wirth氏によってCDC 6000 SCOPのもとに作成された[1]。以後、数回の言語仕様の変更に伴って処理系も改められている。一方で、非CDC機械への移植も試みられてきた。最初のもものは、ICL 1900への移植[2]であり、PASCALで書かれたコンパイラをCDC 6000のコンパイラで翻訳・実行したものである。

最近、日本国内でも[3]で示されているようにいくつかの処理系が作られているが、ここでは、東京大学工学部情報工学専門課程のFACOM 230-38に於けるPASCAL処理系(以後、PASCAL 230と呼ぶ)の作成と問題点等について述べる。

なお、PASCALの処理系として仮想のスタック機械を設計してその機械のシミュレータを作るだけで移植ができるPASCAL-P[4]と呼ばれるものもあるが、ここで述べる処理系は実存する機械の目的プログラムを生成するものである。

2. PASCAL処理系の移植

PASCAL 230は、PASCALコンパイラとPASCALエータとで構成される。[5]

PASCALエータは、FACOM 230 OS II/VSのEとでPASCALコンパイラおよびPASCALコンパイラで生成される目的プログラムを管理する。PASCALコンパイラは、PASCALの目的プログラムの形式であり、PASCALエータのもとでPASCALプログラムを絶対形式の目的プログラムに翻訳する。コンパイラは1パスであり、目的プログラムは生成されたままの形式で、PASCALエータのもとで実行可能である。なお、PASCALコンパイラを武市、PASCALエータを高橋が担当した。

PASCALコンパイラのプログラムは、H.H. Nägeleの"trunk PASCAL"を母体として、FACOM 230-38の命令を生成するためには修正や変更を加えたものである。トランクコンパイラは、コード生成部分のうち特定の機械に依存する部分がコメントで記述されており、エラーメッセージはCDC 6000のPASCAL処理系のものと同様である。[6]

全体のプログラム行数は5800でtrunk PASCALの原稿テキストの1/3はそのままである。

移植は、[2]で述べられている方法と同様に図1のようにして行なった。

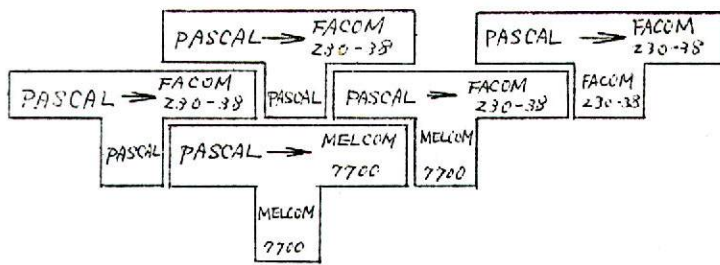


図1. PASCAL 230の移植の過程

プログラムは、最初、東京大学教育用計算センター-MELCOM 7700のPASCAL処理系[3][7] (以後、PASCAL 7700と呼ぶ) でコンパイルを行った。約12回のコンパイル/実行によって、コンパイラ部分のFACOM 230-38機械語が得られた。このデバッグプログラムで発見されたエラーの殆んどは、record structureのvariant partに関連するものであった。これは、PASCAL 7000がselected fieldとtagの値の矛盾性をチェックしていないためであった。なお、配列の添字の値もチェックしていた。

2週間後、バイナリコードがFACOM 230-38に移され、PASCALモータのサポートのもとでFACOM 230-38上でデバッグを行った。ここで12個のエラーが発見されたが、バイナリコードを直接修正する事で対応した。

standard PASCALとPASCAL 230で処理される言語ではいくつかの違いがある。

standard PASCALに付してつぎの事を制限し

(1) type alfaを除いてpacked data typeが無い。

(2) formal parameterとしてprocedure functionは許されない。

つぎの事を拡張してある。

(1) 16ビット表現の整数としてtype shortintが許される。

なお、PASCAL 230システムの大きさは約120Kバイトである。

3. PASCAL 処理系の経験と問題点

PASCAL 230の移植を開始してから約2ヶ月後、研究室内での使用により数件の誤りが発見された。原則として当日中に新しいシステムに変更する作業が行われた。これは研究室内の違った分野に於ける使用で、コンパイラが自身をコンパイルする工で発見できない種類のエラーであった。

PASCAL処理系の定量的な評価は、充分でないが[3]で報告されている。目的プログラムの実行効率はおもかくとして所要記憶容量が比較的大きい。これは、例えば、コンパイラ自体をセグメント分けする方法なども考えられる。

これを改良するとしても既存の処理系を使用する場合、相当量の修正が必要である。記憶容量も含めて効率の良いコンパイラを移植する方法を考える必要がある。

4. 終わりに

PASCALという比較的大きな言語の処理系を容易に作成できた。PASCALコンパイラがPASCAL語で書かれている。したがって処理系の作成に際して、コンパイラが自分自身をコンパイルする事になる。製作者自身が利用者の立場にあり、カレントなシステムに対して利用者の立場に立つアドバイザーであるという事は、ソフトウェアシステムの開発に当って重要な事であると考える。

最後に、本稿の発表の機会を与えられた東京大学和田野教授ほかに當てての田端部長、秋山部長に深謝致します。

参考文献

- [1] N. Wirth: The design of a Pascal compiler, software-practice and Experience 1, pp. 309-333 (1971)
- [2] J. Welsh, C. Quinn: A pascal Compiler for the ICL 1900 Series Computer, Software Practice and Experience 2 pp. 77-77 (1972)
- [3] 武市, 足田, 安村: 東京大学のPASCALコンパイラについて, 情報処理学会第16回大会, (1975)
- [4] K.V. Nori, U. Ammann, K. Jensen, H.H. Nägeli: The Pascal <P> Compiler: Implementation Notes, Berichte der Fachgruppe Computer-Wissenschaften Nr.10, ETH Zürich (1974)
- [5] 武市, 高橋: Pascal 230 処理系概要: 内部資料 (1976)
- [6] K. Jensen, N. Wirth: Pascal - User Manual and Report, Lecture Notes in Computer Science No. 18, Springer-Verlag (1974)
- [7] SFER PASCAL, Le Langage de programmation PASCAL-Compilateur pour les ordinateurs CII 10070, IRIS 80, IRIA (1975)

