7B－5

OS／omicron第４版における Modula−2言語処理系の実現

外川良太郎，森永智之，早川栄一，並木美太郎，高橋延匡

（東京農工大学　工学部）

1.はじめに

現在、我々の研究室では、手書き文字を中心としたマルチメディアデータを取り扱う OS である OS/omicron 第 4 版（以下 V4)を開発している。

V4 では、多種多様なデータを管理するために、二次元アドレス空間やダイナミックリンク機構等を実現機構に採用している。このため、開発用の言語処理系も V4 の実現機構を考慮しなければならない。我々の研究室では、V4 の実現機構を考慮した言語 C 処理系を実現している。だが、実際に使用するうちに、開発に不向きな点も生じた。

本稿では、V4 用 Modula-2[1][2] の実現について述べる。

2. ダイナミックリンク環境における言語 C の問題

ダイナミックリンク環境や 2 次元アドレス空間を考慮した言語 C 処理系を実現する場合、次の問題点が生じる。

(1) リンク時における識別子のサーチ

ダイナミックリンク環境下では実行時に識別子のサーチを行なう。この時、システム内の全ファイルを対象とし、識別子をサーチすると、サーチに余分な手間がかかる。

(2) 分割コンパイル時の識別子の属性

分割コンパイルでシステムを開発すると、ファイルのリンク時に、識別子の定義側と参照側で識別子の型が違うことがある。

そのため、コンパイル時にお互いの情報を知ることができないので、参照側と定義側と識別子が一致しているかどうかの確認がとれない。

これらは、言語 C のスコープ規則と、コンパイル時に呼び出される側のファイルの情報が分からないことが原因である。

3.設計方針

2. で挙げた問題を解決するため、Modula-2 を用いて、V4 用記述言語の開発を行う。

C のスコープ単位は主にファイルであり、識別子のスコープもファイルの内側か外側かの区別しかない。そこで、ファイルとは別の論理的単位、すなわち、モジュールというスコープ単位をもつ Modula-2 を用いて、2.で挙げた問題を解決する。

本稿では、Modula-2 を用いた V4 用言語処理系の設計・開発を行なった。これにあたり、まず、次の設計方針を立てた。

(1) Modula-2 のスコープを V4 の名前空間に対応させる。
(2) Modula-2 の輸出宣言・輸入宣言に対してダイナミックリンク機構を用いる。
(3) Modula-2 の実行環境は V4 の実行コンテキスト[3]を用いる。

4.設計・開発

3. で記述した方針を基に、言語処理系の設計・開発を行なった。

(1) 輸出・輸入宣言とダイナミックリンク

V4 では、外部識別子のアクセスを、リンクテーブルを利用したダイナミックリンクで行なう[4]。

リンクテーブルには、外部識別子の実体の 2 次元アドレス、識別子を含んでいるモジュール名、識別子の型、等の情報が格納されている。外部識別子にアクセスするためには、識別子の 2 次元アドレスを確定しなければならないが、識別子のサーチの手間を軽くするために、リンクテーブル内でサーチするモジュール名を指定する。

また、外部識別子のリンク前にテーブル内の型と識別子実体とを比較することで、呼び出す側と呼び出される側のそれぞれのモジュール内に存在する同名の識別子が一致しているかどうかの確認もとれる。

Implementation of the Programming Language Modula−2 for the OS／omicron Version 4
TOGAWA Ryoutarou , MORINAGA Tomoyuki , HAYAKAWA Eiichi , NAMIKI Mitarou and TAKAHASHI Nobumasa
Faculty of Technology , Tokyo University of Agriculture and Technology

リンクテーブル内のモジュール名や型などの情報は輸出宣言や輸入宣言の情報を基に作成する。

Modula-2 のもつ輸出宣言、輸入宣言とリンクテーブルとの対応を図 1 に示す。

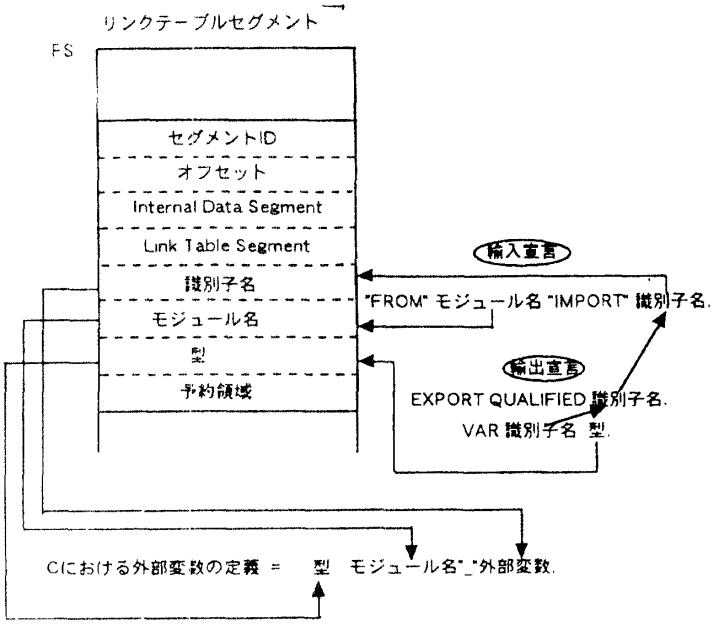

図1 リンクテーブルと輸出入宣言の対応

(2) 入れ子構造に伴う識別子名の変換

Modula-2 では、モジュールや手続きの入れ子を認めているので、同じ名前の識別子が一つのモジュール内に存在できる。そのため、モジュール内に同じ名前の識別子が複数存在しても、それぞれ別の手続きの内部で定義されてれば、Modula-2 の BNF では許される。

これを、C の BNF にすると次のようになる。

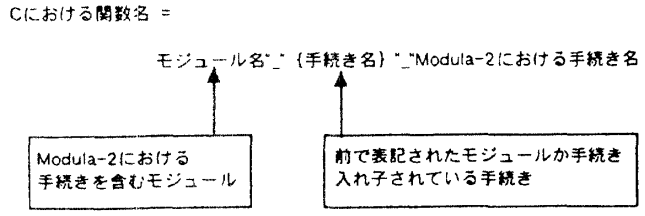

図2 名前の変換

名前を区別するのに、"_" を用いる。言語 C の識別子名では "_" の表記が許されているが、Modula-2 の識別子名には "_" の表記が許されていないからである。

変数を記述する場合は、識別子のスコープを保護するために、スコープを表すディスプレイを言語 C の構造体を使って記述する。これによって、識別子は構造体のメンバとして表すことができる。

(3) 輸出宣言と輸入宣言の変換

呼び出される側のコンパイルでは、定義モジュールの変換が行なわれる。定義モジュール内の識別子が変数であった場合、変数の実体もこの定義モジュールの中でしか宣言されていない。そこで、言語 C への変換時にはこの変数を外部変数としなければならない。

一方、呼び出す側では、輸入宣言の変換が行なわれる。このときにわかる情報は識別子名とモジュール名の情報だけであり、その識別子の種類や型まではわからない。

そこで、この識別子の型や種類の情報を得るために、V4 用 Modula-2 では実現モジュールと同時に、外部変数の実体が定義されている定義モジュールもオープンし、両者の情報を比較する。これにより、識別子の種類と型が一致しているかどうかを確認し、必要ならば外部変数定義や外部変数参照宣言を作成する。

5. おわりに

本稿では、V4 用 Modula-2 の開発について述べた。これにより、ダイナミックリンク時における識別子のサーチをオーバヘッドを減らすことが可能になる。

今後の課題として、識別子のサーチのオーバヘッド量等の性能面に関する評価を行うことである。

参考文献

[1] N.Wirth:PROGRAMMING IN MODULA-2,Springer-verlag Berlin Heidelberg, 1985

[2] Paul M.Chirlian 他:Modula-2 入門, アスキー,1986

[3] 中村他:80386 用 OS/omicron 開発のための言語 C 処理系の実行環境の設計, 情報処理学会第 44 回全国大会, 2f-1, 1992

[4] 森永他:OS/omicron V4 のためのマイクロカーネルの設計, 情報処理学会コンピュータシステムシンポジウム論文集, pp.35-42, 1994