OS／omicron 第４版におけるModula−2 言語処理系の実行環境の設計

3N－3

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、計算機で仮想化した紙をユーザに提供する、手書き入力を支援する OS である OS/omicron 第 4 版（以下 V4）を開発している。この OS では手書きデータなどを管理するため、実現機構として 2 次元アドレスやダイナミックリンクを採用している。

このようなシステムを開発するには、2 次元アドレスやダイナミックリンクに対応した言語処理系が必要になる。かつて、我々はこれらの機構を考慮した言語 C 処理系[2] を開発した。

しかし、開発を進めるうちに言語 C では型チェックや変数のスコープに問題があることがわかった。そこで本稿では、V4 用の Modula-2 [1] 処理系の実行環境の設計について述べる。

2.言語 C 処理系における問題点

V4 の開発を行う際に、言語 C では次のような問題がある。

(1) 分割コンパイル時における型属性

C でシステム開発を行なうとき、通常は分割コンパイルで行なう。そのため、同じ識別子名でも、定義側のファイルと参照側のファイルでは識別子の型属性が違うことがある。しかし、コンパイル時にこのエラーを発見することは困難である。

(2) 識別子のスコープ

C の識別子のスコープ単位はファイルである。そのため、C のスコープは一つのファイル内かシステム全体の 2 種類しかなく、複数のファイルをまとめて一つのスコープを設定することができない。そのため、特定ファイル以外からのアクセスを禁止しているデータの作成が不可能になる。また、データの予定外の破壊の恐れもある。

3.設計方針

2. で挙げた問題を解決する V4 用の処理系の開発するため、次の方針を立てた。

(1) ダイナミックリンクや二次元アドレスに対応するために、セグメンテーションやリンクテーブルを採用する。
(2) コンパイル時に型チェックを行うため、リンクテーブルに識別子の型の情報を入れる。
(3) データごとにアクセスの設定を行なうために、変数のスコープ単位をモジュールにする。

4.設計

3. の設計方針をうけ、V4 用 Modula-2 処理系の実行環境の設計を行った。

(1) 全体構成

我々の研究室で開発された V4 用言語 C 処理系 CAT の実行環境は、複数のセグメントで構成されている。大規模なデータの共有のために、一つの外部変数に対して一つのセグメントを割り当て、必要なリンク情報を名前ごとにリンクテーブルにまとめ、それらを一つのセグメントに格納している。

しかし、これだけではデータごとのアクセス設定は行なえない。そこで、識別子のスコープ単位をモジュールとし、モジュール内の識別子は基本的にモジュール内でしか使えないようにする。

識別子を別のモジュールで使用する場合は、まず識別子を定義しているモジュールによるモジュールの外への識別子の輸出、すなわち、「EXPORT 識別子名;」という宣言が必要である。この宣言をまとめたものが定義モジュールである。そこには、データならば識別子名とその型が、手続きならば手続きのプロトタイプが格納される。

輸出された識別子を使用するときは、モジュールは識別子の輸入、すなわち、「IM

Design of the Runtime Environment 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

PORT 識別子 FROM モジュール;」という宣言が必要となる。

輸出された識別子を輸入することで、別のモジュール内の識別子を使用できる。

そのため、識別子の輸出入には定義モジュールが必要になるのだが、CAT の実行環境ではこの情報を格納する場所がない。

そこで、V4 用の Modula-2 では定義モジュールを格納するセグメントを新たに用意した。その実行環境が図 1 である。

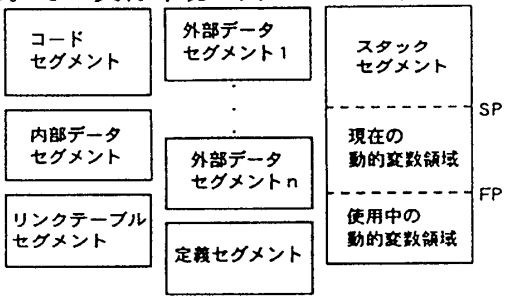

図1 実行環境の全体構成

コンパイル単位はモジュールである。一つのモジュールをコンパイルすると図 1 のようなセグメントの集合になる。

定義モジュール内の情報はすべて一つのセグメント（ 定義セグメント ）に格納する。外部の名前空間だけでは、その識別子が輸出（EXPORT）されているかどうかまではわからないからである。

これによって、リンク時に名前を探すことが容易になり、リンク時間も短縮される。

(2)レジスタ構成

Intel 系のプロセッサでは、セグメント ID をセグメントレジスタに保持しないと、セグメントの参照ができない。その割付けは図 2 のようになる。

レジスタ	指示セグメント
CS	コードセグメント
DS	ワーク用 内部静的変数セグメント
ES	
FS	リンクテーブルセグメント
SS	スタックセグメント

図2 セグメントレジスタの割付け

(3) リンクテーブルの構成

V4 ではリンクテーブルを用いてリンクを行う。リンクテーブルの構成は図 3 のようにした。

「IMPORT モジュール名」とは、名前が格納されているモジュールの名前である。例えば、識別子 a だけでは、モジュール M 内の識別子 a なのか、モジュール N 内の識別子 a なのかわからない。そこで、リンク時には、モジュール名も明確にする。

型情報とは、識別子を参照するモジュールが認識しているその識別子の型属性である。コンパイル時には、この型情報と実体の識別子の型を比較し、型チェックを行なう。型情報には、まず識別子がデータか手続きか、そしてデータならばデータの型が、手続きならば引数の型が記録されている。

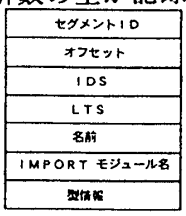

図3 リンクテーブルの構造

(4) ダイナミックリンクの処理

V4 では、外部手続きや外部変数のリンクはダイナミックリンクで行う。リンクの処理過程は従来のダイナミックリンクとほぼ同じである。しかし CAT と違い、Modula-2 ではリンクする識別子、(輸入する識別子)が輸出されているかどうかを確認しなければならない。

そのためリンカは、リンケージフォールトが発生したら、それを起したリンクテーブル内の IMPORT（輸入）モジュール名を参考に、識別子が存在するモジュールの定義モジュールを探し、その定義モジュールから、目的の識別子が輸出されているかどうかを確認する。これは、輸出がされていなければ、その識別子自体が存在していてもリンクはできないからである。

リンクを行なうのは、輸出入が確認されてからである。

5. おわりに

本稿では、V4 用の Modula-2 処理系の実行環境の設計について述べた。今後は詳細設計を進め、処理系を実現する。

参考文献

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

[2] 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-10, 1992

[3] 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