

オブジェクト指向による OS/omicon 第4版の実行環境の設計

6 D - 1

加藤泰志, 早川栄一, 並木美太郎, *高橋延匡

東京農工大学 工学部, *拓殖大学 工学部

1. はじめに

OS/omicon 第4版[1] (以下 V4 と略す) は手書きや画像データといったマルチメディアデータの認識や編集をターゲットとした OS である。マルチメディアデータ処理では、多様なデータが存在し、データの型に応じた資源管理を行う必要がある。これに対して V4 では、マイクロカーネル[2]構成にし、ダイナミックリンク、セグメンテーション、ワンレベルストアなどの機能を提供する。本報告では、V4 に対してオブジェクト指向を導入する場合の実行環境について述べる。

2. V4 用 C++処理系

V4 用の言語処理系としては、現在 C 言語処理系である CAT がある。しかし、C 言語では、マルチメディアデータのような多様なデータに対する型の定義やそのデータを操作するメソッドとの対応関係の管理、拡張、共有のインタフェースなどの点で問題がある。

これらの点からオブジェクト指向が必要となった。そこで、オブジェクト指向言語を使うことにより、クラスを用いてこれらのデータ型とそれを操作するメソッドを定義する。また、オブジェクト指向の機能を用いて、統一的な拡張のインタフェース、共有のインタフェースをユーザに提供する。

オブジェクト指向言語としては C++を使う。オブジェクト指向言語の中で C++を選んだのは、C 言語との互換性、普及していること、柔軟性があるためである。

また、ターゲットマシンとしては、Intel 386 系のプロセッサを使う。これは、V4 ではセグメントが必要なことと、広く普及しているため入手が容易であるためである。

3. V4 用 C++処理系の実行環境

V4 では、多様なデータに対応するため、ダイナミックリンク、セグメンテーション、ワンレベルストアなどの機能を提供している。ここでは、それらの機能と C++の関係について述べる。

また、実行環境の設計方針は

- ・ C++の文法はできるだけ変更しない
- ・ V4 の特徴を生かした設計にする。

とした。

3. 1 セグメントの割当て

図1にソースファイルとクラス定義、セグメントの関係を示す。

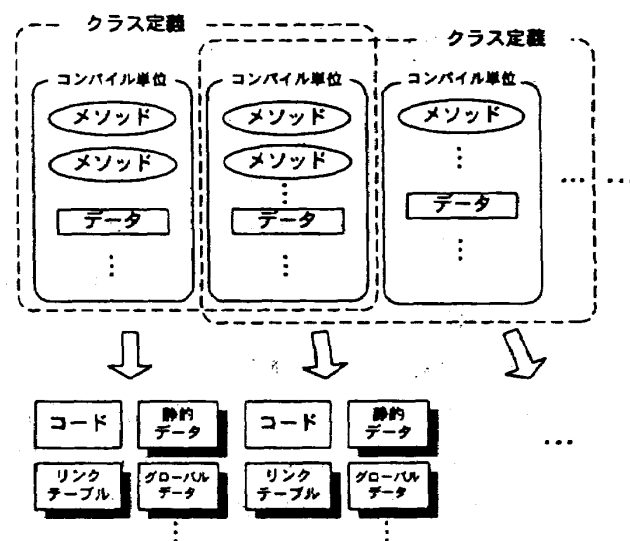


図1 セグメントの構成

静的データセグメントは、静的変数（オブジェクト）を格納する。グローバルデータセグメントはグローバル変数（オブジェクト）を格納する。

また、図の4種類のセグメント以外に実行時にスタックセグメントが割り当てられ、永続オブジェクトを使う場合には、一つの永続オブジェクトに対して一つのセグメントが割り当てられる。

図のリンクテーブル、静的データ、グローバルデータセグメントは、タスク間で共有、非共有どちらにもできる。リンクテーブルをタスクごとに持たせ

Design of a Runtime Environment for the Object Oriented OS/omicon Version 4

Yasushi Kato, Eiichi Hayakawa, Mitarou Namiki and Nobumasa Takahashi

Faculty of Engineering, Tokyo University of Agriculture and Technology

2-24-16, Naka-cho, Koganei, Tokyo 184 JAPAN

た場合、タスクごとに異なるものをリンクすることもできる。また、グローバルデータは、一つのグローバル変数（オブジェクト）に対して、一つのセグメントを割り当てる。これにより、外部データごとに共有、非共有の設定ができる。

3.2 ワンレベルストア、セグメンテーション

V4 では、単一の二次元アドレス空間と、単一階層記憶を持つ。また、セグメントごとに保護ポリシを変更することができ、拡張可能な複数のページを持つことができる。本システムでは、一つの永続オブジェクトに対して、一つのセグメントを割り当てる。これにより、オブジェクト単位での管理、保護、共有が可能となる。また、オブジェクト内、オブジェクト間のリンクをポインタで表現することが可能となる。さらに、V4 ではワンレベルストアであるからオブジェクトをそのまま保存できる。

3.3 ダイナミックリンク

C++の仮想関数のリンクは、レイアウトバイディングと呼ばれ、仮想関数テーブルを使い実行時に呼び出し先が決定される。しかし、実行時にリンクできる範囲は、リンクにより実行ファイルを生成する段階で決まる。

しかし、それでは動的なクラス定義の変更を行うことができない。そこで、本システムでは、実行ファイルを生成せず、仮想関数のリンクにダイナミックリンクを使う。具体的には、仮想関数テーブルをリンクテーブルの中に置き、ダイナミックリンクにより、実行時にリンクを行う。これにより、動的なクラス定義の変更、タスクごとにリンク先を変えることが可能となる。

また、C言語との互換性のために、グローバル関数のリンクにもダイナミックリンクを使う。

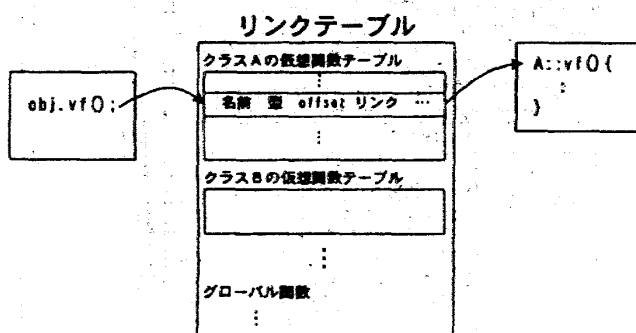


図2 仮想関数のリンク

3.4 クラス・オブジェクトの管理

現在多くのシステムでは、C++が提供するオブジェクト指向の世界は、コンパイラの中で閉じている。

しかしながら、ダイナミックリンクによる動的なクラス定義変更を行うためにはシステムがクラス定義や、オブジェクトに関する情報を知り、その情報を管理する必要がある。

このためこのシステムでは、コンパイラが、クラス定義情報を出力し、それをクラスマネージャが管理する。これらの情報を使い、ダイナミックリンクがダイナミックリンクを行う。

また、オブジェクトマネージャが永続オブジェクトの管理を行う。

3.5 永続オブジェクト

本システムでは、永続オブジェクトをサポートする。永続オブジェクトは、資源としてシステムで管理する。永続オブジェクトの生成、消去、永続オブジェクトへのIDの取り出しは、オブジェクトマネージャにシステムコールを送ることにより行う。

4. おわりに

本報告では、オブジェクト指向によるV4の実行環境の設計について述べた。

現在、V4用C++コンパイラのα版をテスト中である。今後は、コンパイラを完成させて、実際にアプリケーション、ミドルウェアのプログラミングを行い、有効性を検証する。

謝辞

本研究は科学研究費基礎研究 (A)(2)09358004により行われた。

参考文献

- [1] Hayakawa, E., Namiki, M. and Takahashi, N.: Basic Design of SHOSHI Operating System that Supports Handwriting Interfaces, 情報処理学会論文誌, Vol.35, No.12, pp.2590-2601(1994).
- [2] 森永智之, 早川栄一, 並木美太郎, 高橋延匡: 単一二次元アドレス空間を提供する拡張可能なマイクロカーネルの開発, 情報処理学会論文誌, Vol.38, No.5, pp.1016-1025(1997).