ＯＺ：対象指向開放型分散システムアーキテクチャ

－ＯＺ＋のシステム開発環境－

3T-4

塚本享治（電総研）　近藤貴士（シャープ）
水谷功（住友電工）　田中伸明（松下電器）
中込昌吾（ＡＢＣ）

１．はじめに

ＯＺシステムは、分散処理をＯＳＩネットワークで接続された複数の計算機上に存在するオブジェクトの相互作用として捉えた対象指向開放型分散処理システムである。筆者らは、これまでにＯＺシステムを提案し、その実証システムを実現して公開デモを行った。今後、ＯＺ＋システムとして更に拡張を行う予定である。本稿では、ＯＺシステムの実装上明らかになった問題点をあげ、ＯＺ＋システム開発のため実現した開発環境について述べる。

２．ＯＺシステムの開発環境

現在、ＯＺシステムは住友電工Ustationとシャープ0A210、ブロードバンドＬＡＮ、キャリアバンドＬＡＮ、ＯＺブリッジなどから構成されている。各ワークステーションはイーサネットによって接続されている。その開発は次のように行う必要があった。

・移植性の考慮…機種に依存しないソフトウェアの作成。これによって、同一のソースプログラムを複数のマシンで利用する。

・ターゲットマシンごとの開発…実行モジュールの作成、実行、デバグは各ターゲットマシンで行う。

・平行開発…常に同一バージョンの実行モジュールを各マシン上で動作させる。

３．ファイル管理上の問題点とクロスコンパイラ

ソースファイルはバージョン管理等を考えるとある特定のマシンに一元管理されるのが望ましい。しかしターゲットマシンのための実行モジュール生成はそれぞれのマシンでしか行えない。このため現在の環境では各マシン上にソースプログラムのコピーを持つことになり、プログラム修正に伴うバージョン管理が問題になる。また、オブジェクトファイルにはマシン間で共通に利用可能なものがあるが、共有機構を実現しにくい。バークレイ版ＵＮＩＸにはＮＦＳなどの機能があり、マシン間でファイルを共有するのに有効であるが、このメカニズムは現在のところ必ずしも利用できる状況にない。

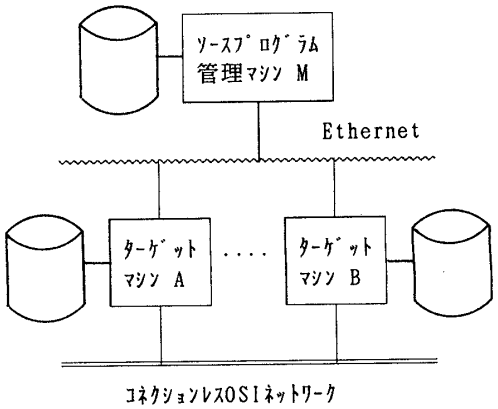


図１．開発機器構成

以上のことから、筆者らはターゲットマシンと開発マシンを分けるべきであると考え、ＳＵＮ３を開発マシンとしUstationおよび0A210をターゲットマシンとしたクロスコンパイラを開発した。これによりソースプログラム管理の簡便化だけでなく、次のことが可能となった。

・開発時の負荷分散…ターゲットマシンへの処理集中を回避できる。

・オブジェクトファイルの共有化…異なるマシン間であってもオブジェクトファイルのレベルで共通に利用できるものがある。

・ターゲットマシンよりも処理能力が高い、あるいは開発環境の整備されたマシンを開発に利用できる。

４．実行モジュールの問題点

ＯＺ＋システムにおいては、ＯＺシステムでは実現されていないネットワーク管理、システム管理、

OZ: Object Oriented Open Distributed System Architecture -- Environment of OZ+ System Implementation --
Michiharu TUKAMOTO(1), Takashi KONDO(2), Shogo NAKAGOME(3), Nobuaki TANAKA(4), Isao MIZUTANI(5)
(1)Electrotechnical Laboratory (2)SHARP Corporation (3)ABC Co.,Ltd.
(4)MATSUSHITA Electoric Industrial Co.,Ltd. (5)SUMITOMO Electric Industries,Ltd.

名称サーバなどを実現する予定であり、これらの記述にはオブジェクト指向が適していると考えている。これらの機能を現在の実行系上で実現するとき、次の問題がある。

（１）モジュールサイズ

現在の実行系の実装では、呼び出される可能性があるプリミティブメソッドはすべて実行系内に前もって準備せねばならない。したがってインタプリタのプログラムサイズは大きくなってしまう。

（２）プログラム修正の難易

わずかな修正を行う場合でも、再コンパイルおよび再リンクにかなり時間を要する。その上、組み込み部分が多く、拡張性に乏しい。

以上の問題を解決するために、立ち上げ時に必要最小限の核部分だけを起動し、付加的な機能は後から付け加えるよう、ダイナミックリンカを開発した。ダイナミックリンカを用いることにより、実行系起動の高速化、システムの軽装化が図れる。また、モジュール（オブジェクトファイル）の共通利用や実行系においてネイティブコードでの実行も可能となる。さらに、実行系の一部の機能しか使用しない、特化したサーバなどの実現が容易になることが予想される。

5．ダイナミックリンカ

一般にＵＮＩＸでは、ダイナミックリンク機構はサポートされていない。今回開発したダイナミックリンカは、ＳＵＮ３およびＳｙｓｔｅｍＶマシン（UstationおよびOA210）で動作する。主な仕様は次の通りである。

・ダイナミックリンクを行う契機は明示的に指定
・リンクされるオブジェクト間の相互参照可能
・シンボルの再定義可能
・関数だけでなく共有変数のダイナミックリンクも可能

ダイナミックリンカはライブラリとして提供される。基本的なインタフェース関数は次の通りである。

・dynamiclink_init(myname)
　… ダイナミックリンクのための初期化を行う。
・dynamiclink(module)
　… moduleをダイナミックリンクする。
・sym2addr(symbol, symval)
　… symbolが解決されたアドレスを求める。

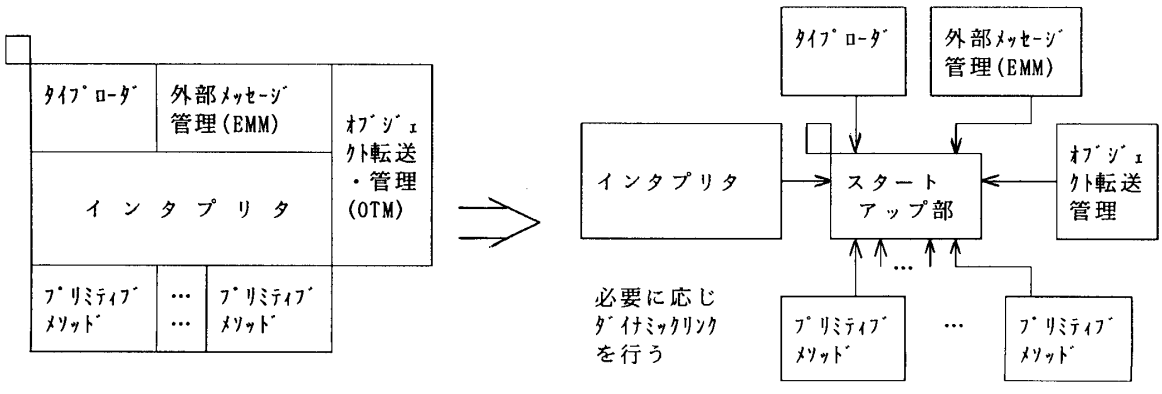
6．まとめ

ＯＺシステムの問題点を開発環境を中心に述べ、改善のため開発したクロスコンパイラとダイナミックリンカについて述べた。今後開発するＯＺ＋システムではこれらを利用した設計・開発を行う予定である。特にダイナミックリンカについては、「軽く、しかも拡張性のある」システムを作成するためのツールとして、今後さらに検討・改良を加えていきたい。以上に述べたクロスコンパイラはＧＮＵプロジェクトのGCC-1.34を、ダイナミックリンカは、comp.sources.unix に投稿された "Dynamic linking package for BSD"（作者Dave Jones氏）をもとに作成した。

なお、本研究は通産省の大型プロジェクト『電子計算機相互運用データベースシステムの研究開発』で行われたものである。

［参考文献］

1）塚本 他,"OZ:対象指向開放型分散ｼｽﾃﾑｱｰｷﾃｸﾁｬ",第38回情処全大,pp.1667-1674 (1989,3)
2）塚本 他,"OZ:ｵﾌﾞｼﾞｪｸﾄ指向開放型分散ｼｽﾃﾑｱｰｷﾃｸﾁｬ - ｵﾌﾞｼﾞｪｸﾄ指向型分散ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ言語とその実装",情報処理学会ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ研究会資料(1989,6)



ＯＺ　　ＯＺ＋

図２．ダイナミックリンクの例