

異種 OS 共存技術 (DARMA) の適用によるシステム移行方式

佐藤雅英*、関口知紀*、木村信二*、新井利明*、富田理**、梅都利和**

* (株)日立製作所システム開発研究所、** (株)日立製作所情報機器事業部

1つのハードウェアプラットフォーム上に複数の OS を共存させる DARMA 技術を適用して、旧システム向け既存ソフトウェアを新システム上で動作させるエミュレータシステムを開発し、銀行端末システムのシステム移行に適用した。開発したエミュレータシステムでは、1 台のターゲットシステム上で、旧システム向け OS 環境と新システム向け OS 環境を共存させた。旧システム向けアプリケーションを、旧システム向け OS 環境で動作させることで、既存ソフトウェアを改造することなく、ターゲットシステム上で実行できる。旧システムから新システムへのシステム移行に対して、開発したエミュレータシステムを用いることにより、システム移行を段階的に実施し、システム移行を実施する際に発生する開発コスト、信頼性低下の問題を緩和した。

System Upgrade Method using the DARMA Technology

Masahide Sato*, Tomoki Sekiguchi*, Shinji Kimura*, Toshiaki Arai*,

Satoru Tomida**, Toshikazu Umedu**

* Systems Development Laboratory, Hitachi, Ltd. ** Mechatronics Systems Division, Hitachi, Ltd.

This paper describes a method for upgrading a system hardware and software using the DARMA technology that executes two operating systems concurrently on a single hardware platform. We developed an emulation of a banking terminal system and applied our upgrade method. The old OS for the banking terminal system and a new operating system coexist on the same platform. The existing application programs are executed on the old OS environment without re-coding, the new OS and its application are used to implement an abstraction of the new hardware for the old system. This method allows upgrades of old applications without recoding, thus keeping upgrades costs very low.

1. はじめに

オペレーティングシステム (OS) の新規開発には多くの工数と時間を要する。そのため、多くの場合、広く普及している事実上の標準的 (de facto standard) な OS を利用してシステムを構築することが多い。

標準的な OS を使用してシステムを構築することの利点は、豊富に流通しているソフトウ

ェアを利用可能であるため、独自開発する部分が少なく、システム開発期間を短縮できる点にある。ISV (Independent Software Vender) は、最もよく売れている OS 上で稼動するソフトウェアを開発することで利益をあげようとしている。この動きは、アプリケーションプログラムだけでなく、周辺装置やそれを制御するドライバ類を開発 / 製品化する場合にも当て

はまる。また、プロセッサチップやボードも標準的な OS を想定したアーキテクチャが早期に開発されるため、ユーザは特に努力することなくハードウェア技術の進歩の恩恵を得ることができる。システムを新規に開発する場合、標準的な OS を使用してシステムを構築することは、開発期間短縮および開発コスト削減に有効である。

しかし、標準的な OS を利用する場合の問題点のひとつに、独自機能の付加が困難である点がある。特に、標準的な OS の内部構造に起因する機能不足の場合は、機能追加がほとんど不可能である。ユーザはこれまでに開発してきた多くのソフトウェア資産を保持していることが多く、そのソフトウェア資産が標準的な OS に対応していない場合には、その資産を捨て、標準 OS に対応したソフトウェアを新たに開発する必要が生じる。たとえば、専用 OS を採用している既存システムを、標準的な OS を採用したシステムに移行する場合、既存システム上で動作するアプリケーションを標準的な OS に対応させる必要がある。しかし、既存のソフトウェアを書き換えることは、工数だけでなく、システムの信頼性を悪化させるという観点からも問題がある。これは、基幹システムや制御システムのような、長期間にわたる開発、保守、運用が必要なシステム構築する際には致命的である。

そこで、上記の問題を解決するために、1 台の計算機上に、複数 OS を搭載する DARMA (Dependable Autonomous Realtime Management) 技術^{1),2)}を適用し、標準的な OS と専用 OS (レガシ OS) を共存させた。既存のソフトウェアをレガシ OS 上で実行させることで、既存システムから新システムへの移行に関して、新旧システムを共存させる中間的なシステムの構築が可能となる。この中間的な

システムを利用することで、段階的にシステムを移行することができる。

2 . オープンシステムへの移行

ネットワーク、パーソナルコンピュータ (PC) の普及に伴い、汎用 PC システムでは、標準的な OS を採用する動き (システムオープン化) が盛んになっている。これは、システム開発時の費用と時間の面から、有利であるためであり、システムの構成部品であるソフトウェアやハードウェアも標準的な OS を想定して開発するベンダが多くなっている。

一方、基幹システムや制御システムのような専用システムでは、専用 OS を採用したシステムが多く存在しており、汎用的な用途を目的に開発された標準的な OS では実現困難な機能や信頼性を実現していることが多い。しかし、この分野のシステムにおいても、標準的なコンポーネントを利用してシステムを構築する動きが盛んになっており、専用システムを継続的に利用することが困難な状況になっている。たとえば、利用しているハードウェアの老朽化、使用ハードウェア部品の供給停止、採用しているソフトウェアのサポート停止、新規ハードウェアサポートのための専用ソフトウェアの開発コスト等の問題のために、既存の専用システムを継続して利用することが困難になり、オープンなシステムへ移行せざるを得ない状況になることも多い。

3 . システム移行時の問題点

専用システムを、標準的な OS を利用しているシステムに移行するためには、以下のような問題がある。

(1) 標準的 OS 利用の問題

標準的な OS を用いたシステムでは、拡張性、信頼性、互換性の問題が発生する³⁾ことがある。

(a) 拡張性、機能追加

標準的な OS は汎用的に開発されているため、特定用途向けの機能が不十分である場合がある。不足機能がアプリケーションプログラムを開発することで補える場合は、新規に開発することで対応できるが、不足機能が OS 構造に依存している場合にはその機能を個別に付加することが不可能である。

(b) 信頼性

標準的 OS およびそのアプリケーションソフトは一般にソース公開されていない。そのため、障害が発生した場合、早急に対応することはほとんど不可能である。また、仕様やソースが開示されている場合でも、障害箇所の特定や修正には不十分であることが多い。さらに、ハードウェア障害の場合には冗長構成や最悪の場合部品の交換によって対策可能であるが、ソフトウェア障害の場合には、開発元が修正するまでは根本的な対策が不可能である場合が多い。以上のことから、基幹システムや公共システム構築の際に標準的な OS を使用する際の制約は大きい。

(c) 従来との互換性

企業の情報システムでは、従来より蓄積されてきた多くのソフトウェア資産がある。これらのソフトウェアは長年にわたって使用されており、十分な信頼性を有している。そのため、標準的な OS 導入のためにこれらソフトウェアを書き換えることは工数だけでなく、システムの信頼性を悪化させるという観点からも問題となる。

(d) ハードウェアコスト

既存の専用 OS を採用したシステムと標準的な OS を採用したシステムでは、各 OS がサポートするハードウェアが異なり、標準的な OS を利用したシステムの移行の場合、上記のソフトウェア書き換えのコストのほかに、ハー

ドウェア交換するためのコストが発生する。一般に、ハードウェアの異なるシステム間の移行の場合、ソフトウェアの開発コストとハードウェアコストが同時期に発生する。

(2) システム切り替え作業時間の問題

新旧システムの切り替え作業をできるだけ短期間で実施する必要がある。企業の基幹システムの移行の場合、移行のために、システム自体を長時間停止することは、ビジネス上も問題となる。

専用システムから、標準的な OS を利用したシステムへ、システムを移行させる場合、以上のような課題を解決する必要がある。

4 . 異種 OS 共存技術 (DARMA) による解決

前章の課題を解決するために、異種 OS 共存技術 (DARMA) を適用し、専用 OS とその OS 上で動作するアプリケーションプログラムを、標準的な OS を利用した計算機システムに共存させた。

DARMA 技術は、1 台の計算機上に複数の OS を搭載させ、OS 間の連携、機能補完によってシステム全体の機能向上を可能とする技術である。DARMA 技術により、必要な機能を高信頼な専用 OS の基に実現することが可能となる。

システム移行に関しても、標準 OS のシステムに、既存の専用 OS を搭載することができ、標準 OS と専用 OS が共存したシステムを構築できる。この共存システムは、システム移行に関して、既存システムと新システムの間隔的なシステムに位置づけることができる。

5 . 異種 OS 共存技術 (DARMA)

DARMA 技術は、ナノカーネル方式による異種 OS 共存技術であり、1 台の計算機上に複数 OS を共存させ、それぞれの OS を同時に動

作させる。DARMA 技術は、標準的な OS を利用しつつ、かつ機能上不可欠な部分には自由に独自の部品を追加できる構造を確保するための基盤として機能する。基本構造と考え方を図 1 に示す。

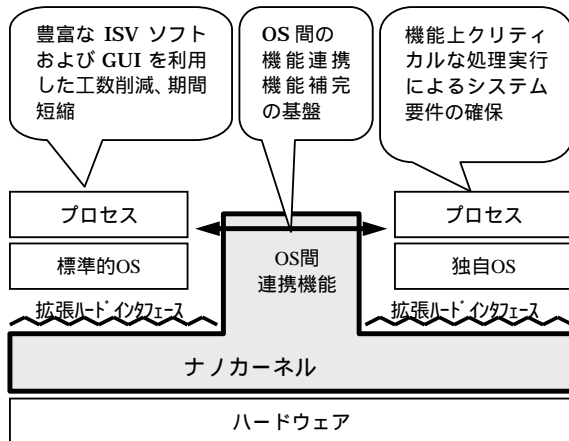


図 1 . ナノカーネルの基本構造

なお、本稿では、DARMA 技術を DARMA ナノカーネル、あるいは単にナノカーネルと記載することがある。

5.1 ナノカーネル機能

ナノカーネルは、表 1 に示した機能の集合で実現している。

表 1 ナノカーネル機能一覧

#	項目	機能概要
1 資源管理機能	メモリ分割	物理メモリを分割し、各 OS、ナノカーネル、共用メモリに割付け
	I/O 装置分割	各 I/O 装置を各々 OS に占有割付け
	プロセッサ分割	プロセッサを時分割して OS 間で共用
	タイマ共用	各プロセッサに仮想タイマを提供
5 OS 間通信機能	共用メモリ	OS 間で共用するメモリを構築
	メッセージパッシング	異種 OS 上のプロセス間のメッセージパッシング
	プロセス間同期	異種 OS 上のプロセス間の同期制御
8	障害監視、回復	障害発生を他 OS に通知 障害 OS 以外の OS の継続実行を保証

(1) 資源管理機能

OS が管理する計算機資源には、プロセッサ、メモリ、I/O 装置がある。資源管理機能は以下のようにこれら資源を分割して各 OS に割り付ける機能である。

(a) メモリ分割機能

物理メモリを分割し、各 OS およびナノカーネル用に割り付ける。また、後述する共用メモリ用のメモリも確保する。各 OS は自分自身に割り付けられたメモリ以外はハード的に搭載されていないものとして実行する。

(b) I/O 分割機能

OS 毎に各々が制御する I/O 装置を占有させる。メモリと同様に、各 OS は自分自身に割り付けられた I/O 装置以外は、ハード的に搭載されていないものとして実行する。

(c) プロセッサ分割機能

プロセッサは各 OS に時分割に割り付ける。割付のアルゴリズムおよび優先順位は共存する OS の特性に依存する。たとえば、汎用的な OS とリアルタイム OS を共存させる場合には、汎用 OS の処理を中断させて、リアルタイム OS に制御を移す必要がある。

(d) タイマ共用機能

タイマは、通常ハードウェアに一つのみ実装される装置であり、OS 間で共用する必要がある。タイマは指定された時間に割り込みを発生させることが基本機能である。ナノカーネルは、各 OS からの割り込み要求を受け取り、各要求をマージして最も近い割り込み時間を物理タイマにセットする。タイマ割り込みが発生した際には、その割り込みがどの OS からの要求に対応するものであるか判定し、該当する OS のタイマ割り込みハンドラへ制御を渡す。(図 2)

(2) OS 間通信機能

ナノカーネルは、共存させた OS 間の機能連携、機能補完を可能とするため、共用メモリ、

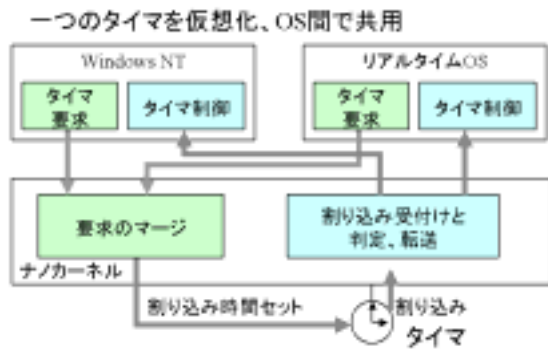


図 2 . タイマ共用機能

メッセージパッシング、プロセス同期の OS 間通信機を提供する。(図 3)

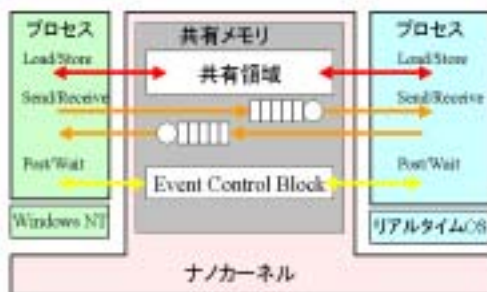


図 3 OS 間通信機能

(a) 共用メモリ機能

OS 間で共通に参照するメモリ領域を設定する機能である。上記のメモリ分割機能で確保した共用メモリ専用物理メモリを利用して各 OS からのアクセスを可能とする。ただし、プロテクション強化のため物理メモリを直接参照することは禁止し、仮想メモリ経由でのアクセスのみを許可する。

(b) OS 間メッセージパッシング機能

異なる OS 上のプロセス間でメッセージパッシングによってデータの共用、転送、プロセスの同期制御を実施する機能である。上記の共用メモリ機能を用いて実装し、共用メモリ上にメッセージキューと送受信に関する制御テーブルを配置する。制御テーブルの排他制御用のロックも制御テーブル上に構築する。

(c) OS 間プロセス同期機能

異なる OS 上のプロセスの同期制御機能である。この機能を使用することにより、各 OS 上のプロセスは、自 OS 上のプロセスと同様に、他 OS 上のプロセスと同期をとりながら処理を実施することが可能となる。

(3) 障害監視、回復機能

OS に障害が発生した際に、その状況を他 OS から監視し、必要に応じて回復処理を起動できる。

以上の機能により、ナノカーネルは、一つのマシン上で複数の OS が共存して動作することを可能としている。

6 . システム移行への適用

上記 DARMA 技術を、銀行端末システムのシステム移行に適用した。銀行端末システムは、長年にわたる開発、運用、保守が必要な基幹システムのひとつである。

6 . 1 既存の銀行端末システムの特徴

ここで、とりあげる銀行端末システムは、以下のような特徴をもっていた。

(1) 銀行端末向け専用 OS

銀行端末システム向けに機能を拡張した専用 OS を採用していた。システムとして必要な機能は、アプリケーションだけでなく、OS レベルの機能も追加して、システムが構築されている。

(2) 豊富なソフトウェア資産

既に、10 年以上使用してきており、専用 OS 向けに開発してきた、高信頼なソフトウェア資産が多く存在していた。また、既存のソフトウェアには、上記専用 OS および当該ハードウェアを前提とした銀行端末システム向け処理が実現されていた。

(3) 古いハードウェアを利用

上記で述べたように、10 年以上前のハード

ウェアを前提に構築されたシステムであり、最新のハードウェアを利用できない。たとえば、I/O 装置についても、専用の I/O アダプタのみが接続可能であり、PCI (Peripheral Component Interconnect) バスや ISA (Industry Standard Architecture) バスを有していない。

6.2 システム移行の必要性

既存のシステムは、機能上あるいは性能上特に問題がない。しかし、既存システムは長年にわたり、運用、保守されてきたために、ハードウェアの継続供給が、部品調達、サポート等の問題により難しくなってきた。そのため、今後のシステムの開発、保守の観点から、標準的な OS を利用した汎用 PC をベースとしたオープンシステムへの移行が必要となった。

6.3 ターゲットシステムの特徴

ターゲットシステム(移行後システム)は以下の特徴をもつ。

(1) 標準的な汎用 OS

標準的な OS として、Windows NT®を採用した。当該 OS は PC および PC サーバ分野では広範囲に利用されており、標準 OS としてふさわしい OS である。

(2) 標準的なハードウェア

PC/AT 互換のハードウェアを利用する。当該ハードウェアも PC / PC サーバ分野で広範囲に使用されており、標準ハードウェアとしてふさわしいハードウェアである。

6.4 システム移行の問題

当該銀行端末システムを、上記のターゲットシステムへ移行する方法として、もっとも単純な方法は、ターゲットシステム上に、既存システムで実現していた機能を移植することである。我々はこの移行方式を一斉移行方式と呼ぶ。しかし、今回の銀行端末システムの移行に関して、一斉移行方式を採用した場合には、以下の

ような問題が発生する。

(1) 既存ソフトウェア資産の移植コスト

既存のソフトウェア資産を、ターゲットシステムに対応させるために書き換えるコストである。ターゲットシステムの OS、デバイスへの対応が必要となる。当該銀行端末システムでは、これまで開発してきた多くのソフトウェア資産があり、これを一斉にターゲットシステム向けに移植するためには、長期の開発期間が必要であり、開発コストも膨大である。

(2) 信頼性低下

既存ソフトウェアの移植のための書き換えは、信頼性にも影響する。基幹システムのソフトウェアは一般に長年にわたる運用、保守を通して高信頼性を達成している。これらのソフトウェアの書き換えは、信頼性確保の観点から問題となる。

我々は、前章で説明した DARMA 技術を適用し、システム移行を段階的に実施する方式を実施した。この方式を段階的移行方式と呼ぶ。

6.5 段階的移行方式

当該銀行端末システムのシステム移行に関して、一斉移行方式を採用する場合の課題を緩和するために、DARMA 技術を適用して、既存システムからターゲットシステムに移行する間の中間的なシステムとしてエミュレータシステムを開発した。その構成を図 4 に示す。ここでは、汎用 OS である Windows NT を銀行端末システムに適用することを目的に、共存する OS として、既存の銀行端末システム専用 OS を搭載した。

このエミュレータシステムでは、ターゲットシステム上で、既存の銀行端末システムの環境をエミュレートする。既存の銀行端末処理を専用 OS 側で実行し、新規に開発するアプリケーション、あるいは、移植が完了したアプリケーションは、Windows NT 側で実行することが

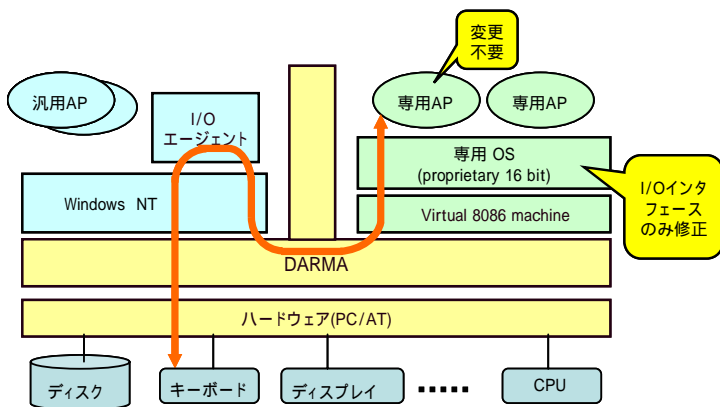


図4 . エミュレータシステム

できる。

(1) 既存ソフトウェア資産移植コスト削減

当該エミュレータシステムでは、汎用 OS である Windows NT と既存の銀行端末システム向け専用 OS を共存させることで、銀行端末システム向けアプリケーションは変更せずに実行することができる。また、専用システムの I/O 要求は、Windows NT 側に、銀行端末向け専用 OS の I/O をエミュレートするための I/O エージェントを設けることで、ターゲットシステムに接続されたデバイスを用いて実行した。これらにより、ソフトウェア資産である銀行端末システム向け専用アプリケーションを変更することなしに、ターゲットシステム上で、実行することができ、ソフトウェア資産の移植コストを削減できる。

(2) 信頼性低下

また、ソフトウェア書き換えによる信頼性低下の問題についても、既存の高信頼なソフトウェアを改造なしに搭載することができ、ソフトウェア書き換えによる信頼性低下を抑えることができる。

エミュレータシステムを活用することにより、既存システムからターゲットシステムへのシステム移行は、2段階に実施可能となる。

以上説明した、段階的移行方式を採用するこ

とで、当該銀行端末システムの移行では、第1段階として、オープンなハードウェア上に、既存システムを少ない改造で搭載することができ、さらに第2段階として、専用 OS 側で搭載している既存ソフトウェアを、徐々に、Windows NT 側に移植していくことができる。

7 . おわりに

既存の銀行端末システムを標準的な OS を採用したオープンシステムに移行させる際の問題を緩和するために、DARMA 技術を適用した。DARMA 技術を適用することで、既存のソフトウェア資産を活かしつつ、システム移行を段階的に実施した。

Windows NT® は 米 国 Microsoft Corporation の米国およびその他の国における登録商標です。

PC/AT は、米国 International Business Machines Corporation の商品名称です。

参考文献

- 1)新井他,「ナノカーネル方式による異種 OS 共存技術「DARMA」の提案」, 情報処理学会, 第 59 回全国大会,講演論文集(1), pp.139-140, 1999
- 2)佐藤他,「ナノカーネル方式による異種 OS 共存技術「DARMA」の実装」,情報処理学会, 第 59 回全国大会,講演論文集(1), pp.141-142, 1999
- 3)「Windows 活用の実態と可能性」,日経デジタルエンジニアリング,日経 BP 社,1998 年 11 月号(1998,No10)