

## 解説



## オープンアーキテクチャの実現に向けて† —PowerPC Platform を例として—

緒 方正 暢†† Mark Dean ††† 三井 信 雄††††

### 1. はじめに

現在、世の中で使われているコンピュータシステムは、バッテリーで動作し自由にもち運べる携帯型システムから1部屋を占有する大きさの大型システムまでさまざまある。このコンピュータシステムの中で最大の比率を占めているのは、個人使用のコンピュータ（いわゆるパソコン）である。これには、IBM PC/AT 互換機や Apple Macintosh などが含まれる。これらのシステムは主に、ワードプロセッサ、スプレッドシート、データベースなどのアプリケーションプログラムを利用するために使用されている。パソコンは、グラフィカルユーザインタフェースの進歩とともに、コンピュータの専門知識をもたない多くのユーザに使用されるようになってきた。今後はますます複雑なグラフィック処理、音声や動画といったマルチメディアデータ処理を活用したアプリケーションプログラムが重要になるため、システム性能に対する要求は衰えることはない。

今後とも、高度化するユーザの要求に応えられるシステムを提供し続けるためには、最新の半導体、インタフェース、記憶装置、表示装置などの技術を取り入れ、それらを有効に使いこなすことができなければならない。そのためには、ハードウェアやソフトウェアの設計が独立して、かつ、柔軟にできる新しいコンピュータアーキテクチャが必要となっている。もちろん、従来のアプリケ

ーション資産を継承しながら、新しい技術が積極的に採用できなければならない。

本稿では、コンピュータシステムが直面しているいくつかの問題点について述べ、PowerPC Platform を例にして今後のコンピュータアーキテクチャのあり方について述べる。本稿でいうシステムアーキテクチャとは、ハードウェアとソフトウェアの境界を定めるインタフェースである。

### 2. 従来のシステムアーキテクチャ

#### 2.1 初期のシステム

メインフレームやオフィスコンピュータのような初期のシステムでは、新たなハードウェアが開発されるたびに、オペレーティングシステムもそのハードウェアに合わせて新たに設計実装されていた。ハードウェアとオペレーティングシステムは密接な関係にあり、アプリケーションプログラムも必要に応じてそのシステムに合わせて開発されていた。

#### 2.2 ワークステーションと UNIX

RISC プロセッサの登場とともに急成長してきたワークステーションは、高度な処理能力、ネットワーク機能などを特徴にしていた。各社から販売されているワークステーションはプロセッサやバスがまったく異なるハードウェア構成ながら UNIX オペレーティングシステムという共通のオペレーティングシステムが稼働するという特徴をもっている点が従来のシステムと大きく異なっている。

UNIX オペレーティングシステム自体は異なるハードウェアシステム上で動作させるための高い移植性（ポータビリティ）をもっているのが特徴である。最も初期の UNIX オペレーティングシステムではコンピュータの研究者向けに配布されたため、カーネル、システムツールなどの全ソースコードが無償で添付されており、コンパイラ

† Towards the Open System Architecture—A Case Study of the PowerPC Platform Architecture— by Masanobu OGATA (PowerPC System Design Center, IBM Japan, Ltd.), Mark Dean and Nobuo MII (Power Personal Systems Division, IBM Corp.).

†† 日本アイ・ビー・エム(株)事業企画・推進、システム・デザイン・センター

††† IBM コーポレーション

†††† IBM コーポレーション、現在米国 SegaSoft

\* 本文中に出てくる製品名等は各社の登録商標もしくは商標です。

も提供されていた。カーネルのサイズも小さく、ハードウェアに適合するようにオペレーティングシステムそのものを変更し、再コンパイルするといった移植作業はコンピュータの専門知識をもっている研究者には比較的容易であった。また、デバイスを追加するたびにカーネルを再構築しなければならなかったが、今日のようにデバイスの種類も少なく、機能も限られていたので、カーネルの再構築はさほど困難な作業ではなかった。UNIX 上のアプリケーションはソースコードで提供されており、たとえ異なるハードウェア上であっても共通のプログラミングインタフェースを提供する UNIX 上では高い移植性が保てた。

### 2.3 マイクロカーネル

本来、小さくて移植性の高かった UNIX オペレーティングシステムも、ネットワーク機能など多くの機能を取り込んだため肥大化してしまっただ。また、マルチプロセッサシステムのような新しいシステムアーキテクチャが登場し、従来のカーネルのままでは対応できず、カーネルそのものを根本的に見直す必要がでてきた。オペレーティングシステムはハードウェアを直接操作するため、ハードウェアとの依存関係が大きい。オペレーティングシステムの中でハードウェアに依存した部分はごく一部に限られている。そこで、ハードウェアに依存しカーネルの基本的なサービスを提供する部分と、ファイルシステムやネットワーク機能といったオペレーティングシステム機能を提供する部分に分離することができる。このようにオペレーティングシステムの小さなカーネルの基本部分をマイクロカーネルと呼ぶ<sup>1)</sup>。

マイクロカーネルを採用したオペレーティング

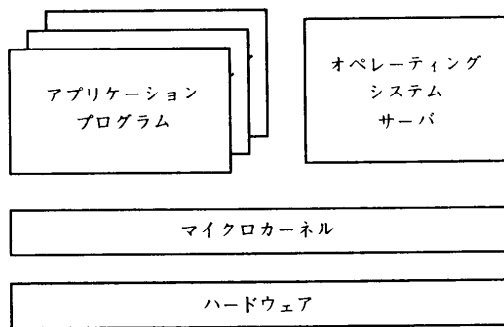


図-1 マイクロカーネルを使ったオペレーティングシステムの実装

システムとして、OSF の OSF/1、IBM の OS/2 Warp Connect PowerPC Edition (ここでは、単に OS/2 for PowerPC と呼ぶ)<sup>2)</sup>がある。OS/2 for PowerPC は、PowerPC を搭載したシステムで稼働する IBM マイクロカーネル上に OS/2 のオペレーティングシステムサービスを実現したものである。IBM マイクロカーネルは、仮想記憶管理、タスクおよびスレッド管理、プロセス間通信、I/O 管理、割込み管理の基本機能のみを行っている。つまり、マイクロカーネルだけがプロセッサ固有およびハードウェアに依存する情報を管理している。ほかのオペレーティングシステムの機能は、サーバとして実装されている (図-1)。これによって IBM マイクロカーネル上のデバイスドライバを除いたサーバ群はハードウェアプラットフォームへの依存性がなくなる。異なるハードウェアに対応させるには、OS/2 for PowerPC のマイクロカーネル部分だけをそれぞれのハードウェアに合わせて開発すればすむ。

### 2.4 パーソナルコンピュータ

マイクロプロセッサの登場とともに成長してきたパーソナルコンピュータシステムは、当初は各社ごとに異なるシステムアーキテクチャだったので、専用のオペレーティングシステムが必要だった。ところが、IBM PC/AT の仕様が一般に公開され、互換機が多数登場し、チップセットや部品の単価が下がるとともにハードウェアの共通化が進んだ。共通化されたハードウェア環境を前提にしたソフトウェアの開発が可能になり、DOS や Windows といったオペレーティングシステムやワードプロセッサやスプレッドシートのようなアプリケーションプログラムが登場した。これらは、ハードウェアシステムとは独立に販売されるようになった。これがいわゆる、「シュリンクラップ」という形態で、実行形式のファイルのみで販売提供されている。利用者はソフトウェアを導入するだけですぐに使えるので、コンピュータに対する深い専門知識はとくに必要なくなった。Windows の登場でアプリケーションプログラムのハードウェアプラットフォームに対する依存度はさらに減少し、大きな市場を形成した。ただし、これらのソフトウェアは特定のプロセッサ、および、ハードウェア構成に密接に依存した実行形式のプログラムであるため、新しいハードウェア

アに柔軟に適應することができない。

### 2.5 ハードウェアアブストラクション

マイクロソフト社の Windows NT は、マルチプロセッサシステムの対応や、Intel 系プロセッサのほかに、MIPS, Alpha, PowerPC のような RISC 系プロセッサを搭載したシステムへの対応を行っている。そのため、HAL (Hardware Abstraction Layer) というオペレーティングシステムとハードウェアの中間のソフトウェア層を設けた。

HAL は OS のハードウェアに依存する部分のコードをオペレーティングシステムから分離し、明確なインタフェースのもとでアブストラクション層として集めたものである。オペレーティングシステムは、HAL を介して、プロセッサ、システムレジスタ、割込みコントローラ、入出力装置などを操作する。HAL はオペレーティングシステムを再構築することなくハードウェアに合わせて交換できるようになっている。こうすることで、新しいプラットフォームへの対応は HAL の変更だけですみ、オペレーティングシステムのバイナリ互換を保つことができるようになった。HAL はマイクロカーネルのようなカーネル機能そのものではなく、ハードウェア依存のサービスを提供するソフトウェア層である。

## 3. 次世代システムアーキテクチャ

### 3.1 従来システムの問題点

パソコンでは当初 DOS の基本的な機能を BIOS と呼ばれる ROM 上のサービスルーチンに分割することでハードウェアの違いを吸収しようとした。しかし、直接ハードウェアを操作する DOS アプリケーションも多く、BIOS だけでソフトウェアの互換性を保つことはできなかった。また、DOS のために本来設計された BIOS はほかのオペレーティングシステムにとっては機能が十分でないため、あまり利用されることはなかった。

今日の Windows のようなオペレーティングシステムには標準でウィンドウシステムや各種のサービスを提供するソフトウェアが標準に含まれている。そのため、オペレーティングシステムが提供するプログラミングインタフェースだけを使ったプログラミングが主流になり、かつてのように

直接ハードウェアを操作するアプリケーションプログラムはほとんどなくなった。プログラムの互換性はオペレーティングシステムで保たれるようになったが、基本ソフトウェアの量は膨大になっている。

しかし、あらゆるハードウェアを想定してオペレーティングシステムをあらかじめ作っておくことは困難であるため、新しいハードウェアが登場するたびに、変更する必要がある。しかも、このように巨大化したオペレーティングシステムを移植する手間は大きい。マイクロカーネル技術を採用することでオペレーティングシステムの移植の手間を小さくすることができるが、依然、ハードウェアシステムごとにカーネルを変更しなければならないことにはかわりがない。Windows NT の HAL ではシステムごとに合わせてオペレーティングシステムの変更を動的に対応させることができるようにした。しかし、HAL は Windows NT だけのためのソフトウェアであり、ハードウェアプラットフォームメカは HAL を作成してもほかのオペレーティングシステムには対応できない。

したがって、現在のパソコンでは、標準の DOS や Windows がそのまま動作するようにハードウェアを設計することが行われている。しかし、そのためには、チップレベルでのハードウェアの互換性が要求される。つまり、同一の操作レジスタをもつ互換デバイスを使用しなければソフトウェアの互換性を保つことができない。そうすると、ハードウェアメカはオペレーティングシステムがすでに稼働しているハードウェアをコピーするしかない。現在のアーキテクチャに性能や機能の面で限界があることが分かっている。ソフトウェアの互換性を維持するために時代遅れになったハードウェアを使い続けなければならない。これは、最新のハードウェア技術を積極的に採用することを阻害し、ハードウェアの改良、進歩を抑制する原因となっている。

### 3.2 次世代システムアーキテクチャへの指針

- オペレーティングシステムやアプリケーションプログラムのバイナリ互換性を損なわずに、システム設計者が新しい機能を自由に付加できること。つまり、ハードウェアの差異を吸収する仕組みがあること。

- ハードウェアとソフトウェアの設計が独立して、かつ、柔軟にできること。つまり、ハードウェア設計者が提供しなければならない機能とソフトウェア設計者が仮定できる機能が明確に定義されていること。
- 各種のシステム構成に対応できること。たとえば、携帯性に優れた小型システムとマルチプロセッサ構成のサーバのようなシステムではメモリ容量やデバイスの数も異なり、オペレーティングシステムの構成を変える必要がある。アドレス空間の使い方が柔軟に変えられたり、割込み処理が統一的に行えることが重要である。また、マルチプロセッサシステムでもシングルプロセッサシステムでも対応できるようにする。
- 今後登場する新しいシステムに対応できること。たとえば、64ビットプロセッサを搭載したシステムが登場しても、新たにソフトウェアを書き換える必要がなく、従来の32ビットのソフトウェアをそのまま、64ビットプロセッサ上でも利用できること。
- 1種類のオペレーティングシステムであらゆるアプリケーションプログラムに適した環境を提供することは困難であるので、複数のオペレーティングシステムが1つのハードウェア上で選択して使えること。そのためには、選択ブートの方法、ディスクエリアの定義が必要である。また、ビッグエンディアンのオペレーティングシステムでもリトルエンディアンのオペレーティングシステムでも使えるようにするためには、エンディアンの初期化の方法を定義しておく必要がある。
- エラー処理、エラー報告の共通の手段が提供されていること。ハードウェアのエラーに応じた処理をソフトウェアで対処できなければならない。

### 3.3 実現手法

この新しいコンピュータアーキテクチャの実現の鍵は、互換性や性能を損なわずソフトウェアからのハードウェアの抽象化を行うことである。オペレーティングシステムから見たハードウェアのインタフェースを明確に規定し、ハードウェアの差異をアブストラクションソフトウェアで吸収する(図-2)。

電源が投入されると、システムテストとデバイ

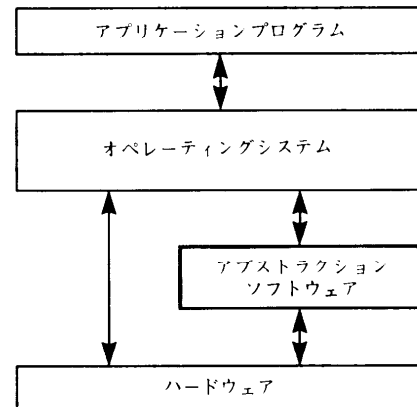


図-2 アブストラクションソフトウェアを使ったオペレーティングシステムの実装

スの検出および初期化を実行するのがブートファームウェアである。これは、ROM内に格納されハードウェアシステムとともに提供される。また、このファームウェアはオペレーティングシステムのブートも行う。このように、ファームウェアは起動時のハードウェアとオペレーティングシステムとのインタフェースを担当している。以下、ブート時に必要な機能を列挙する。

- オペレーティングシステムを起動するためには、ブートデバイスをアクセスするためのドライバ、および、複数のオペレーティングシステムの基本的なブート手順の定義が必要となる。マルチプロセッサシステムの場合、プロセッサの始動方法はとくに重要な問題である。
- 起動時のエラーメッセージはどうやって出力するかも考慮しなければならない。なぜなら、出力装置を自由に変えるためには出力デバイス自身を仮想化しておかなければならない。
- 各種のハードウェアの構成に適するようにオペレーティングシステムを設定するには、システムに関する十分な情報が得られることが必要である。

オペレーティングシステムがブートされた後もハードウェアを抽象化するソフトウェアが必要である。実行時アブストラクションソフトウェアは、プロセッサ操作、キャッシュ操作、メモリ管理、I/O操作、割込み処理など、オペレーティングシステムのハードウェア依存部分を抽象化する。このソフトウェアはハードウェアの設計者によって提供され、あらゆるオペレーティングシ

テムに対応できなければならない。

このようなアブストラクションソフトウェアをハードウェア設計者が提供することで、オペレーティングシステムの互換性を保ったまま、自由にハードウェアを変えることができるようになる。

#### 4. 実 現 例

ここでは、1994年に公開されたPowerPC Reference Platform (PReP)<sup>3)</sup>、および、その後継で1995年に公開されたPowerPC Common Hardware Reference Platform (CHRP)<sup>4)</sup>仕様を例として紹介する。これらのシステム仕様はPowerPC<sup>5)</sup>プロセッサを搭載したコンピュータシステムを対象にしている。本稿では、この仕様をPowerPC Platformアーキテクチャと呼ぶ。PowerPC Platformアーキテクチャでは、ソフトウェアの互換性を維持しながら高性能なハードウェア技術を柔軟に取り入れることを可能にしたアブストラクションソフトウェアを導入した。アブストラクションソフトウェアを導入することで、ハードウェアプラットフォームメーカ、デバイスアダプタメーカ、オペレーティングシステムメーカにとっては、相互の依存関係を最小にできる利点がある。ハードウェアプラットフォームに依存する部分はアブストラクションソフトウェアによって吸収され、アプリケーションおよび大部分のオペレーティングシステムのコードの共有化がはかれるようになる。さらに、ハードウェアプラットフォーム、デバイスアダプタ、オペレーティングシステムのそれぞれの互換性が維持することができるようになる。これは、パーソナルコンピュータの標準化にとって重要な点である。

PowerPC Platformアーキテクチャでは以下の目標を目指した。図-3にPowerPC Platformの構成図を示す。

- ポータブル機からサーバまで基本的に1つのアーキテクチャで対応する。
- 1つのハードウェアで複数のオペレーティングシステムを動作させる。
- 業界標準の仕様を積極的に取り入れ、最新の半導体、装置などを積極的に利用できるようにする。
- システム構成に適したアドレスマップの再配置を行えるようにし、その情報はブート時にオペ

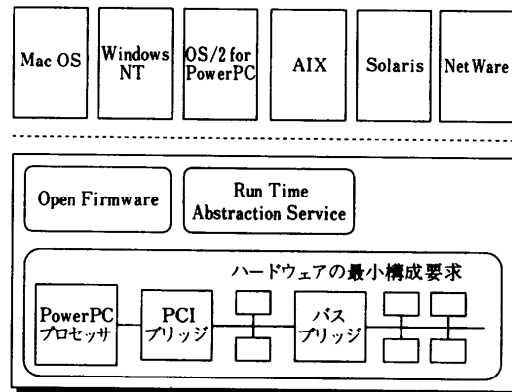


図-3 PowerPC Platformの構成

レーティングシステムへ渡される。

- マルチプロセッサシステムやパワーマネージメント機能のような新しい技術に対応する。
- 将来の機能拡張を可能にする。

##### 4.1 Open Firmware

PowerPC PlatformアーキテクチャではIEEE標準のOpen Firmware<sup>6)</sup>を採用した。Open Firmwareは、ハードウェアの初期化、設定を行いその情報をデバイスツリーとして管理する。さらに、エンディアンモードの設定を行い、オペレーティングシステムをロードする。

Open Firmwareでは、プログラミング言語Forthで生成したFcodeと呼ぶバイトコードプログラムを使ってデバイスアダプタの初期化を行う。Fcodeはブート時に本体上のForthインタプリタによって逐次実行され、プロセッサに依存しないコードになっている。

##### 4.2 実行時アブストラクションサービス (RTAS)

PowerPC Platformアーキテクチャでは実行時のアブストラクションソフトウェアとしてRTAS (Run Time Abstraction Service)を新たに規定した。RTASを利用するための情報は起動時にオープンファームウェア経由でオペレーティングシステムへ渡される。RTASはハードウェアとともに供給され、すべてのオペレーティングシステムから利用することができる。

#### 5. おわりに

PowerPC Platformアーキテクチャでは、アブストラクションソフトウェアの導入により、ソ

ソフトウェアの互換性を維持しながらハードウェア技術の進歩をすみやかにコンピュータに取り入れることを目指している。アプリケーションプログラム、および、大部分のオペレーティングシステムのコードのバイナリ互換性を向上し、アプリケーションも含めてソフトウェアの開発期間の短縮と過去の蓄積された資産の継承を可能にしている。

### 参 考 文 献

- 1) Accetta, M. et al.: Mach: A New Kernel Foundation for UNIX Development, 1986 USENIX Conference, pp. 93-112 (1986).
- 2) Christopher, K. et al.: OS/2 Warp Connect PowerPC Edition, IDG Books, ISBN: 1-56884-458-1 (1995).
- 3) IBM: PowerPC Reference Platform Specification Version 1.1, IBM Corporation (1993).
- 4) Microprocessor Common Hardware Reference Platform: A System Architecture, Morgan Kaufmann Publishers, Inc., ISBN: 1-55860-394-8 (1995).
- 5) IBM: The PowerPC Architecture, A Specification for a New Family of RISC Processors, Morgan Kaufmann Publishers, Inc., ISBN: 1-55860-316-6 (1994).
- 6) IEEE: 1275-1994 Standard for Boot (Initialization Configuration) Firmware, Core Requirements and Practices (1994).  
(平成7年7月4日受付)



緒方 正暢 (正会員)

1961年生。1984年広島大学工学部第二類(電気系)卒業。1986年同大学院システム工学専攻博士課程前期修了。同年日本アイ・ビー・エム(株)入社。東京基礎研究所にて分散実時間オペレーティングシステム、マイクロカーネル、モバイルコンピューティングに関する研究に従事。現在、同社開発製造部門PowerPCシステム・デザイン・センター勤務。電子情報通信学会、IEEE、日本ソフトウェア科学会各会員。

### Mark Dean

Mark Dean received a BSEE degree from the University of Tennessee in 1979, a MSEE degree from Florida Atlantic University in 1982 and a Ph. D. in Electrical Engineering from Stanford University in 1992. He is an IBM Fellow responsible for architecture and performance for the RS/6000 Division. He was the director of architecture for the Power Personal Systems Division responsible for creating the PowerPC Reference Platform specification.



三井 信雄 (正会員)

1931年生。1955年九州大学工学部通信工学科卒業。同年日本放送協会入局。経営情報室副主管を経て、1969年日本アイ・ビー・エム(株)入社。1990年同社副社長開発製造部門担当。同年IBMコーポレーション副社長兼任。1991年IBMコーポレーション副社長兼エントリ・システム・テクノロジー担当ゼネラルマネージャ。1995年より米国SegaSoft会長兼CEO。情報処理学会役員(1979-80)。通産省工業技術院日本工業標準調査会基本問題懇親会委員(1990-91)。工学院産業技術審議会委員(1991)。日本産業技術振興協会、電子情報通信学会各会員。