

# Keio-MMP における

7R-8

## マイクロカーネルアーキテクチャ†

和田 英彦<sup>1‡</sup> 河内谷 清久仁<sup>2‡</sup> 緒方 正暢<sup>2‡</sup> 西尾 信彦<sup>3</sup> 追川 修一<sup>3</sup> 徳田 英幸<sup>3</sup>

<sup>1</sup>横河電機(株) <sup>2</sup>日本アイ・ビー・エム(株) <sup>3</sup>慶應義塾大学

### 1 はじめに

慶應義塾大学では、分散マルチメディア統合環境プロジェクト(Keio-MMPプロジェクト)を行なっている。本稿では、Keio-MMPにおけるソフトウェアアーキテクチャ[1]の中で、基本ソフトウェアプラットフォームであるマイクロカーネルに必要とされる機能について述べるとともに、現在本プロジェクトで行なわれているカーネルの改良について報告する。

### 2 Keio-MMPにおけるマイクロカーネル

本プロジェクトでは、動画や音声等のように比較的大量でアクセスに対して時間的制約を持つ連続メディアデータを、分散した環境で扱う。連続メディアの処理は、CPUの高速化、メモリの大容量化のようなハードウェア技術や、データ圧縮のようなソフトウェア技術等の進歩に伴い、一般のワークステーションやパーソナルコンピュータでも実行が可能となってきた。しかし、UNIX<sup>1</sup>に代表される従来のオペレーティングシステム上で分散マルチメディアアプリケーションを実行すると、タイムシェアリングの概念に基づいたラウンドロビンスケジューリングやFIFOを用いたリソース管理による処理や通信の予測不可能な遅れにより、動画表示が一時的に停止したり、音声途切れてしまうなど、連続メディアを正しく処理できない場合が生じる[2]。また、連続メディア処理を行なわない一般のアプリケーションとの共存も難しい。このような問題を解決して、連続メディア統合環境を構築するためには、リアルタイム処理技術が重要となる。以上の理由から、本プロジェクトではカーネギーメロン大学で開発されているリアルタイムマイクロカーネルのReal-Time Mach3.0(RT-Mach)[3]を採用している[4]。

#### 2.1 連続メディア処理とリアルタイム OS

連続メディアを処理する場合に考えなければいけない、リアルタイム処理技術の問題としては、

- リアルタイムスケジューリング問題
- リアルタイム同期問題
- リアルタイム通信問題

がある[5]。

ここでは、スケジューリング/同期問題について論じる。スケジューリング問題について、例えば、画像、音声などの連続メディアオブジェクトをワークステーション上で多重処理する場合を考えると、あるプロセスの連続メディアデータの処理中に、他のユーザプロセスの起動などによってプロセッサが横取りされてしまうと、画像や音声の再生が正しく行なえないことがある。また、周期的なアクティビティの制御や一時的な過負

荷状況への対応等が、従来のタイムシェアリングのスケジューリング機能では困難である。

リアルタイム同期問題については、プライオリティインバージョン[6]の問題がある。プライオリティインバージョンは、タスク間の同期に関してだけでなく、横取りできないサービスを提供しているサーバ、ネットワーク・アクセスプロトコルなど、あらゆるシステムレベルで起きうる基本的な問題である。また、マルチメディアシステムにおいては、“リップ・シンク問題<sup>2</sup>”のような、二つのデータストリーム間での同期と、ビデオフレーム間での同期のような一つのデータストリーム内での同期の両方の問題がある。

以上の問題に対応するために、RT-Machに用意されているリアルタイム処理技術として、リアルタイムスレッド、リアルタイムスケジューラ、リアルタイム同期機構、リアルタイムプロセス間通信などの機能がある。

#### 2.2 連続メディア処理とマイクロカーネル

従来のUNIX等のオペレーティングシステムでは、単層構造をしたカーネルがその機能を提供したが、オペレーティングシステム自体が肥大化し、機能の変更や保守、管理が困難になった。一方RT-Machは、マイクロカーネルの構成であり、スレッドやIPC等いくつかの基本的な機能のみをカーネルが提供し、それ以外のオペレーティングシステムの機能(例えばファイルシステム)を、ユーザレベルのサーバで提供でき、機能の変更などが容易になった。しかし、マイクロカーネルの構成をとった場合、オペレーティングシステムとして提供する高レベルの機能をどのレベルでサポートするか検討が重要となる。

マイクロカーネル構成を取った場合に、オペレーティングシステムの機能をどのレベルで提供するかにについて次の3つの構成が考えられる[7](図1)。

1. マイクロカーネル
2. ユーザレベルのサーバ
3. アプリケーションのアドレススペース

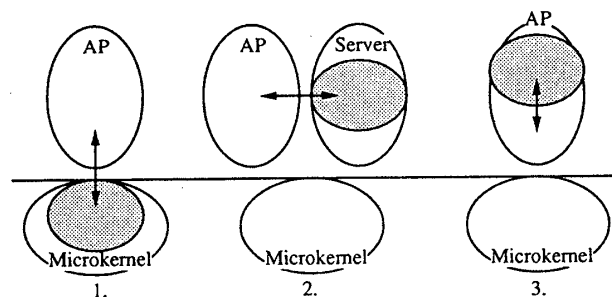


図1: システム構成

この場合に3.の構成で実現した場合には、それぞれのアプリケーションに適した形で必要な機能を実現でき、さらにカーネ

†“Micro-kernel Architecture in the Keio-MMP Project”†

Hidehiko Wada<sup>1</sup>, Kiyokuni Kawachiya<sup>2</sup>, Masanobu Ogata<sup>2</sup>, Nobuhiko Nishio<sup>3</sup>, Shuichi Oikawa<sup>3</sup>, Hideyuki Tokuda<sup>3</sup>

<sup>1</sup>Yokogawa Electric Co., 2-9-32 Nakacho, Musashino-shi, Tokyo 180, Japan, <sup>2</sup>IBM Research, Tokyo Research Laboratory, <sup>3</sup>Keio University

†この研究は、情報処理振興事業協会(IPA)が実施している開放型基盤ソフトウェア研究開発評価事業「マルチメディア統合環境基盤ソフトウェア」プロジェクトのもとに行なわれた。

‡開放型基盤ソフトウェア湘南藤沢キャンパス研究室の研究者としてIPAに登録されている。

<sup>1</sup>UNIXはX/Openの登録商標です。

<sup>2</sup>話している人の唇の動きと音声を同期させること。

ルやサーバとの通信を減らすことができるので、性能向上が期待できる。

また、マルチメディア処理を考えた場合に、違った特徴を持つ複数のメディアを扱う必要がある、それぞれのメディアの特性が異なることが予想され、個々のアプリケーションレベルで柔軟な処理ができることが必要である。このような要求を満たすためには、どのような処理をしたいかというポリシーと、それを実現するためのメカニズムが分離されていた方がよい。このようなポリシーとメカニズムを分離して提供するためには、メカニズムをユーザレベルで自由に実現することができるようなマイクロカーネルの構成を取っていることが望ましい。

### 3 Keio-MMP におけるカーネルの改良

分散連続メディア統合環境を実現するために、RT-Mach に実時間スレッドモデルの拡張 [8]、ユーザレベル実時間スレッドパッケージとユーザレベルスケジューラ [9]、連続メディアオブジェクトに対応したメモリ管理機構 [10]、ユーザレベル I/O ドライバ [11] などの改良を行なっている [4]。今後さらに以下の改良が考えられる。

#### 1. システム負荷情報の提供

システムがどの程度の負荷状態であるか、現在はリアルタイムスレッドがそのデッドラインをミスすることにより判断しており、システムがどのような負荷状態であるかを知る手段は提供されていない。システムの負荷状態をユーザレベルから知ることができれば、QOS 制御のためのサーバ [12] などが、より適切な QOS 制御を行なうことが可能になる。

#### 2. 資源の予約

マルチメディアシステムでは、複数のプログラムが、限られた資源、例えば CPU を協調的に利用する。そのために、プログラム側から必要な資源を予約してもらい、資源の利用を保證できる機構を提供することが必要となる [13]。資源予約機構を実現することで、高いプライオリティを持つ連続メディア処理のプログラムが、割り込み処理によって時間制約が満たせなくなることを防ぐことができる。また、CPU を占有してしまうような動作をするプログラムに対しても、CPU 占有状態を回避することができる。このような資源予約の機構を提供する場合には、プログラム側から出される要求が、システムとして受け入れ可能かどうかを判断するアドミッション制御が重要となる。

#### 3. 特権ユーザ

現在の RT-Mach では、リアルタイム機能関連のシステムコールを発行するには、特権ユーザ (UNIX のスーパーユーザ) の権利が必要である。これは、認証サーバなどと組み合わせることによって、より柔軟な形で機能を提供する必要がある。

### 4 関連研究

ケンブリッジ大学のペガサス (Pegasus) プロジェクト [14] では、オペレーティングシステムとして、Nemesis と呼ばれるマイクロカーネルを使っている。このカーネルは、複数の保護ドメインを持つ 64 ビットの単一アドレス空間をサポートすることで、ドメイン間のデータを共有する構造を簡単にしたり、論理アドレスキャッシュを使用した場合のコンテキストスイッチ時に発生するコストを削減している。また、ユーザレベルスレッドスケジューラを導入したり、キャッシュをいくつかのパーティションに分割して、連続メディアのデータをダイレクトに挿入することによって処理の効率を上げようとする研究 [15] も行なっている。

また、ランカスター大学では連続メディアアプリケーションを処理するために、商用マイクロカーネルである Chorus を拡張する研究を行なっている [16]。この研究では、通信、スケジューリングとデバイス管理を統合することによってストリームベースの通信の実現や、通信路を経由しても QOS を制御できることを目指している。

### 5 おわりに

本稿では、分散マルチメディア統合環境の実現のための基本ソフトウェアプラットフォームとして、リアルタイムマイクロカーネルである RT-Mach を採用し、さらに機能拡張や改良を行なっていることを述べた。

今後は、拡張した RT-Mach をベースにして、Conductor/Performer アーキテクチャ [17] の実現などミドルウェアとの統合を図る。また我々のアーキテクチャは、モバイル計算機や組み込み型システムにも適用できることを実証していく。

### 謝辞

本研究を進めるにあたり、有益なコメントを頂きました慶応義塾大学の斎藤信男教授と「マルチメディア統合環境基盤ソフトウェアプロジェクト」のメンバの方々に深く感謝致します。

### 参考文献

- [1] 徳田, 萩野, 斎藤: “分散マルチメディア統合環境 Keio-MMP プロジェクトにおける連続メディア処理のためのソフトウェアアーキテクチャ,” 第 49 回情報処大全論文集 (7R-1)(1994).
- [2] 中島, 徳田: “マルチメディアシステムのためのオペレーティングシステム技術,” 信学会「マルチメディア技術とその応用」講習会資料, pp.29-36(1993).
- [3] H. Tokuda, et al.: “Real-Time Mach: Towards a Predictable Real-Time System,” USENIX Mach Workshop, pp.73-82 (1990).
- [4] 西尾, 他: “マイクロカーネルによる連続メディア処理の基盤技術,” 第 5 回コンピュータシステムシンポジウム論文集, pp.17-24 (1993).
- [5] 徳田, 斎藤: “マルチメディア統合環境プロジェクトにおけるリアルタイム処理技術,” 情処研報, Vol.93, No.20, 93-ARC-99, pp.9-15(1993).
- [6] L. Sha, et al.: “Priority inheritance protocols: An approach to real-time synchronization,” Technical Report CMU-CS-87-181, Carnegie Mellon University, November(1987).
- [7] H. Tokuda, et al.: “Towards a New Operating System Architecture: Micro Kernel vs. Reflective Architecture,” 日本ソフトウェア学会全国大会 (1992).
- [8] 河内谷, 緒方, 徳田: “連続メディア処理のためのリアルタイムスレッドモデルの拡張,” 第 48 回情報処大全論文集 (1H-1), pp.4-17-4-18(1994).
- [9] 追川, 徳田: “連続メディア処理のためのスレッドメカニズム,” 第 49 回情報処大全論文集 (7R-6)(1994).
- [10] 盛合, 木原, 南部: “連続メディアオブジェクトに対応したメモリ管理機構について,” 情処研報, 94-OS-64, pp.73-78(1994).
- [11] 持田, 他: “Real-Time Mach 3.0 における I/O サーバの構成と評価,” 第 48 回情報処大全論文集 (1H-2), pp.4-19-4-20(1994).
- [12] 河内谷, 緒方, 徳田: “Real-Time Mach 上での QOS 制御サーバの実験,” 第 47 回情報処大全論文集 (4V-3), pp.2-355-2-356(1993).
- [13] C. Mercer, et al.: “Processor Capacity Reserves: An Abstraction for Managing Processor Usage,” Proc. of Workshop on Workstation Operating Systems, October(1993).
- [14] S. Mullender, et al.: “Operating-System Support for Distributed Multimedia,” Proc. of 1994 Summer USENIX, pp.209-219, Boston(1994).
- [15] D. McAuley: “Operating-System Support for the Desk Area Network,” Proc. of the 4th Int'l Workshop on Network and Operating System Support for Digital Audio and Video, pp.3-10, November(1993).
- [16] G. Coulson, et al.: “Extending the Chorus Micro-Kernel to Support Continuous Media Applications,” Proc. of the 4th Int'l Workshop on Network and Operating System Support for Digital Audio and Video, pp.49-60, November(1993).
- [17] 西尾, 他: “Keio-MMP における Conductor/Performer アーキテクチャの協調性能評価,” 第 49 回情報処大全論文集 (7R-7)(1994).