システムソフトウェアと オペレーティング・システム 74－26
マルチメディア通信と分散処理 81－26
（1997. 2. 28）

アイソクロナス・スケジューラの設計と性能評価

竹内 理　岩嵜正明　中原雅彦　中野隆裕　芹沢 一
（株）日立製作所システム開発研究所

近年、VODシステムなどの連続メディア処理を行なうアプリケーションの実現要求が高まっている。我々は、上記アプリケーションの実現に適したマイクロカーネルHiTactixを開発した。HiTactixは、連続メディア処理を行なうスレッドを厳密に一定の間隔で周期駆動可能にするアイソクロナス・スケジューラを備える。さらに、優先度逆転の発生時間を最小限に抑える排他制御機構、スレッド間同期機構を備える。また、非同期イベント処理を階層構造の割り込みハンドラで行ない、連続メディア処理と非同期イベント処理を共存可能にしている。本稿では、それらの概要及びアイソクロナス・スケジューラの周期駆動性能の定量的な評価結果につき述べる。

A Design and Performance Evaluation of Isochronous Scheduler

Tadashi Takeuchi　Masaaki Iwasaki　Masahiko Nakahara　Takahiro Nakano
Kazuyoshi Serizawa
Systems Development Laboratory, Hitachi Ltd.

Recently, necessity of applications to process continuous media data such as VOD systems is becoming high. We developed a micro-kernel named HiTactix which is suitable for implementing these applications. HiTactix provides *isochronous scheduler* that schedules precisely periodically threads which process continuous media data. It also provides an exclusion control mechanism and a synchronization mechanism which minimize duration periods of priority inversions. Besides, it provides multiple layered interrupt handlers which process sporadic events. These layered interrupt handlers enable continuous media and sporadic events to be processed at the same time. This paper gives outlines of HiTactix real-time scheduling mechanisms described above. This paper also shows evaluation results of *isochronous scheduler*'s real-time scheduling ability.

1 はじめに

VOD(Video On Demand)システムやマルチメディアCSCW(Computer Supported Cooperative Work)を典型例とする、連続メディア処理（動画、音声などの連続メディアデータの圧縮・伸長や入出力処理）を行なうアプリケーションが、ここ2～3年実用化に向けて急激な展開を見せている[7]。

連続メディア処理を行なうスレッド（以後、周期スレッドと略す）は、厳密に一定の周期で周期動作する[6][注1]。そのため、連続メディア処理の実現には、スレッドの周期駆動を保証するインタフェースを備えるスケジューラをオペレーティング・システムが提供する必要がある。さらに、優先度逆転(Priority Inversion)を最小限に抑える排他制御機構とスレッド間同期機構、及び、パケット到達などの非同期イベントが多発した場合にも[注2]、周期スレッドの周期駆動間隔のゆらぎを最小限に抑える非同期イベント処理方式を提供する必要がある[5]。

上記課題を解決するために、Earliest Deadline First [1]、Priority Ceiling Protocol[3]、Deferrable Server[4]などのリアルタイム・スケジューリング・アルゴリズムが提唱されてきたが、いずれもその処理オーバーヘッドが大きく、高性能な連続メディア処理アプリケーションの実現を困難にしている。

我々は、上記課題を低オーバーヘッドで解決する連続メディア処理向きマイクロカーネルHiTactixの研究開発を進めている。HiTactixは、周期スレッドの周期駆動を保証するアイソクロナス・スケジューラを備える。さらに、細粒度プリエンプト制御による排他制御機構、スレッド・グループによるスレッド間同期機構、階層構造をした割り込みハンドラによる非同期イベント処理機構を提供している。これにより、資源競合や非同期イベントの多発時にも周期スレッドの駆動間隔のゆらぎを最小限に抑えることが可能になっている。本稿では、それらの概要とアイソクロナス・スケジューラの周期駆動性能[注3]の定量的な評価結果について述べる。

2 HiTactixのリアルタイム・スケジューリング機構の概要

本章では、HiTactixのリアルタイム・スケジューリング機構の概要として、周期スレッドの周期駆動を保証するアイソクロナス・スケジューラ、及び優先度逆転解消方式、非同期イベント処理方式につき述べる。

注1 例えば、動画データを画面に表示する場合、画面出力を行なうスレッドを厳密に一定の間隔で周期駆動しなければ、出力結果がぎこちなくなってしまう。

注2 CSCWなどでは、連続メディアデータの再生処理と並行して、大量の連続メディアデータの受信処理を行なう必要がある。

注3 本稿では、周期駆動性能を、周期スレッドの周期駆動間隔のゆらぎの大きさにより評価する。

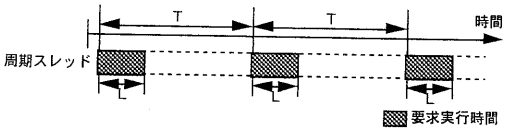

図1 周期と1周期あたりの実行時間の要求

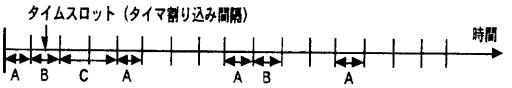

図2 タイムスロット・テーブルの作成

表1 周期と1周期あたりの実行時間の要求例

	T	L
A	40ms	10ms
B	80ms	10ms
C	160ms	10ms

2.1 周期駆動の保証

連続メディア処理は、一定の処理内容を長時間にわたり周期的に繰り返す。そのため、その実行を開始する前に、予めその処理を行なう周期スレッドの駆動周期Tと、1周期あたりの実行時間Lを予測可能である。アイソクロナス・スケジューラは、各周期スレッドに対し、その実行開始前に必要なTとL（図1参照）を宣言することを要求する。

スケジューラは宣言を受け付けると、各スレッドが要求したTとLを満たすタイムスロット・テーブルを作成する。タイムスロット・テーブルとは、図2に示すように、タイマ割り込み発生時刻を境界に時間を分割し（以後、この分割単位をタイムスロットと呼ぶ）、各タイムスロットごとにスケジューリングすべき周期スレッドを記したテーブルである。

このタイムスロット・テーブルは機械的に作成可能である。例えば、周期スレッドA～Cが、表1に示すTとLを要求した場合、これらの要求を満たすタイムスロット・テーブルは図2のようになる。

タイムスロット・テーブルの作成を完了したアイソクロナス・スケジューラは、図3に示す様にタイマ割り込みを契機に起動し、タイムスロット・テーブルに記された周期スレッドを順番にスケジューリングする。

アイソクロナス・スケジューラは、各周期スレッドのためのCPU時間を予め一定周期ごとに確保する。そのため、ある周期スレッドの実行が、他の周期スレッドの実行により妨げられないことを保証できる。また本スケジューラは、周期動作を要求しない通常のスレッド（以後、通常スレッドと略す）を、周期スレッドが割り当てられていないタイムスロットを用いてラウンド・ロビン方式でスケジューリングする。そのため、周期スレッドの実行が、通常スレッドの実行により妨げられないこと

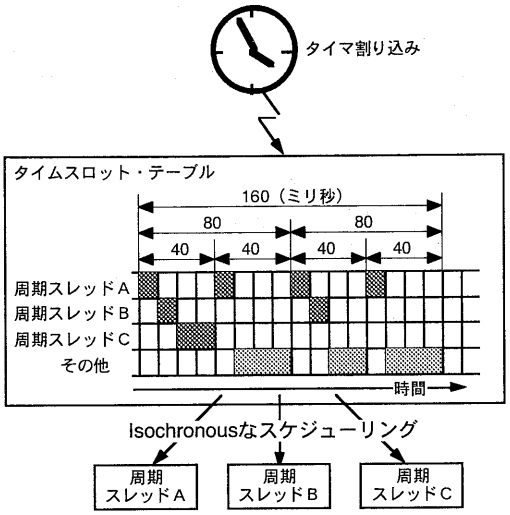

図3　周期スレッドの起動

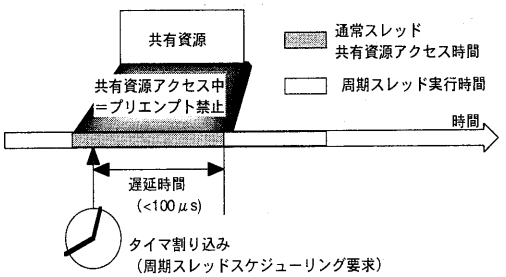

図4　細粒度プリエンプト制御

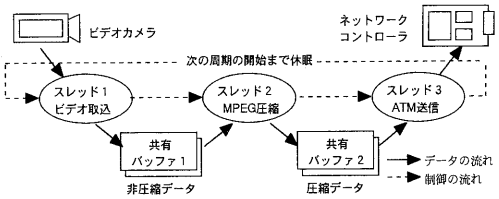

図5　スレッド・グループの使用例

も保証できる。すなわち、本スケジューラは、一定間隔で確保したタイムスロットにおいて必ず周期スレッドがRun状態に遷移することを保証する。従って、各周期スレッドの駆動間隔は厳密に一定になる。

2.2　優先度逆転の解消

長時間にわたる優先度逆転の継続は周期スレッドの実行遅延をもたらし、その結果周期駆動性能の低下を招く。優先度逆転の発生継続時間の極小化のため、HiTactixは、細粒度プリエンプト制御機構とスレッド・グループ機構を提供する。本節では、それらの概要を述べる。

2.2.1　細粒度プリエンプト制御

HiTactixでは、カーネル内資源の排他アクセス制御を細粒度のプリエンプト制御により実現する[5]。これにより、図4に示す様に、周期スレッドのスケジューリング要求がアイソクロナス・スケジューラにより発行されてから、実際にスケジューリングされるまでの遅延時間を100マイクロ秒以内に抑えている。

2.2.2　スレッド・グループ

HiTactixは、ユーザ資源の排他アクセス制御、周期スレッド間の同期機構の実現のために、スレッド・グループを提供する。すなわち、HiTactixでは、単一の周期スレッドだけでなく、複数の周期スレッドから構成されるスレッド・グループに対するタイムスロットの割り当てをアイソクロナス・スケジューラに要求可能にしている。スレッド・グループに属する各周期スレッドは、割り当てられたCPU時間内であれば、同一グループに属する他の周期スレッドを自由にハンドオフ・スケジューリングできる。

例えば、図5に示す様な一連のパイプライン処理を行なう場合、スレッド1～3を一つのスレッド・グループとし、スレッド・グループに対する周期的なCPU時間の割り当てをスケジューラに要求する。

スケジューラは、要求されたCPU時間を確保した後、周期的にスレッド1を起動する[注4]。以後、スレッド1及びスレッド2は、1周期分の実行を完了した後に、次にスケジューリングされるスレッド（スレッド2及びスレッド3）をハンドオフ・スケジューリングする。スレッド3は、1周期分の実行を完了した際に、CPUの解放をスケジューラに対して宣言し、スレッド1～3は次の周期まで休眠する。

スレッド・グループの提供によりストリーム間同期の実現が容易になる[注5]。また、図5の共有バッファなどの、周期スレッド間で共有するユーザ資源に対するスレッドのアクセス順序を予測可能にし、ユーザ資源の排他アクセス制御を不要にしている[注6]。

[注4] スレッド・グループ内で最初にスケジューリングすべきスレッドをグループマスタ・スレッドと呼ぶ。グループマスタ・スレッドは、スケジューラにCPU時間の割り当てを要求する際に併せて指定する。

[注5] 例えば動画ストリームと音声ストリームの同期を実現したい場合、動画処理スレッドと音声処理スレッドでスレッド・グループを形成し、スレッド・グループに対する周期的なCPU時間の割り当てをスケジューラに要求する。以後、動画処理スレッドと音声処理スレッドの1周期分の実行が交互に行なわれ、ストリーム間同期が実現する。

[注6] 例えば図5の共有バッファ1は、スレッド1がまずアクセスし、次に、スレッド1によりハンドオフ・スケジューリングされるスレッド2がアクセスする。即ち、スレッド1が共有バッファ1にアクセス中、スレッド2が共有バッファ1にアクセスすることはあり得ない。

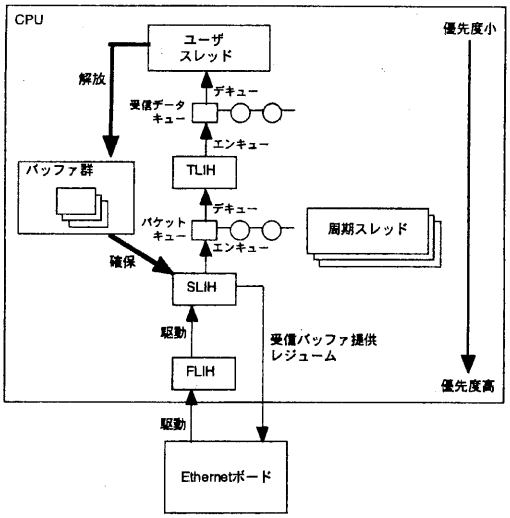

図6　パケット受信用割り込みハンドラの構成

2.3　非同期イベント処理

HiTactixは、パケット到達などの非同期イベントが多発する状況においても、周期スレッドの周期駆動を保証するため、非同期イベント処理を階層構造をした割り込みハンドラにより行なう。すなわち、割り込みハンドラをFLIH、SLIH、TLIH[注7]に分離し、FLIH、SLIHを周期スレッドより高優先度で、TLIHを周期スレッドより低優先度で実行する。FLIH、SLIHの処理を極小化し、割り込み処理の実行による周期スレッドの実行の遅延を最小限に抑えている。

Ethernetボードからのパケット受信処理を例に上記を説明する。パケット受信を行なう割り込みハンドラの構成を図6に示す。FLIHは、パケット到達によりEthernetボードが発行する割り込みを契機に起動する。FLIHは、CPUの割り込み管理機構が提供する割り込みハンドラで動作し、SLIHの起動要求をスケジューラに対して発行する。スケジューラは、実行中のスレッドがカーネル内のプリエンプト禁止区間を走行中の場合のみ、そのスレッドがプリエンプト禁止区間の走行を完了するまでSLIHの起動を遅延する。SLIHは、Ethernetボードが受信したパケットのキューイング、及びEthernetボードに対する新しいバッファの供給、Ethernetボードのレジューム処理を行なう。またTLIHは、SLIHがキューイングしたパケットのデキュー、UDP/IP(またはTCP/IP)などのプロトコル処理、及びユーザ・スレッドに渡すべき受信データのエンキューを行なう。

FLIH/SLIHを周期スレッドより高優先度で起動することにより、パケット到達からEthernetボードのレジューム処理までの時間を最小限にし、Ethernetボードのパケット受信の失敗によるパケット喪失の発生を最小限にしている。また、プロトコル処理を周期スレッドより低優先度であるTLIHで行なうことにより、パケット受信処理(FLIH/SLIHの実行）に伴なう周期スレッドの実行の遅延も小さくしている[注8]。

3　アイソクロナス・スケジューラの周期駆動性能の評価

本章では、前章で示したリアルタイム・スケジューリング機構を備えるHiTactix上で動作する、アイソクロナス・スケジューラの周期駆動性能の定量的な性能評価結果を示す。

3.1　測定条件

本稿で示す各測定の測定条件を以下に示す。

使用マシン	PC-AT互換機 (Pentium 133MHz 搭載)
使用Ethernet ボード	DEC DC21040 (10Mbps Ethernet Card)

測定は、キャッシュ、TLBを全パージした直後に行なった。そのため、以下に示すスケジューリング・コストや周期駆動のジッタ・サイズは、そのワースト・ケース時の値を表している。また、スケジューリング・コストに関しては、ベスト・ケース時の値も測定した。

3.2　スケジューリング・コストの評価

Earliest Deadline Firstをはじめとする従来のリアルタイム・スケジューリング方式は、そのスケジューリング・コストが大きいことが問題となっている。アイソクロナス・スケジューラは、各周期スレッドの実行開始前にタイムスロット・テーブルを静的に作成し、スケジューリング・コストの削減をはかっている。本節では、アイソクロナス・スケジューラのスケジューリング・コストの測定結果を示す。

アイソクロナス・スケジューラのスケジューリング・コストは、スレッド・スイッチのコストに加えて、

1）タイムスロット・テーブル参照による、次にスケジューリングすべきスレッドの決定
2）スケジューリングすべき周期スレッドのスレッド制御ブロック内データの更新
3）スケジューリングすべき周期スレッドの優先度変更
4）スケジューリングされていた周期スレッドの優先度変更

注7 FLIH/SLIH/TLIHはFirst/Second/Third Level Interrupt Handlerの略

注8 TLIHは通常スレッドの中では最高優先度で実行される。また、アイソクロナス・スケジューラは、タイムスロット・テーブルを作成する際に、一定周期(160ms)につき一定時間(10ms)にわたり、いかなる周期スレッドの割り付けをも行なわないタイムスロットを確保している。そのため、パケット到達から160ms以内に、必ずTLIHが駆動する。

表3　最長プリエンプト禁止区間（マイクロ秒）

proc_create	46.8	thread_create	40.3
proc_delete	38.1	thread_delete	48.5
proc_resume	22.1	thread_resume	13.0
proc_suspend	23.5	thread_suspend	23.9
proc_get_info	4.2	thread_get_info	8.9
proc_set_info	3.1	thread_set_info	7.7
proc_get_my_pid	0.0	thread_start_cyclic_exec	43.5
vm_allocate_region	51.5	thread_stop_cyclic_exec	35.0
vm_share_region	42.5	thread_raise_handoff	30.4
vm_deallocate_region	49.6	thread_create_thread_group	27.5
pm_allocate_phys_page_set	55.9	thread_delete_thread_group	20.9
pm_share_phys_page_set	38.4	thread_get_my_tid	0.0
pm_deallocate_phys_page_set	33.1	event_create_ebox	15.3
pm_map_phys_page_set	50.6	event_delete_ebox	10.2
pm_unmap_phys_page_set	44.3	event_init_and_ebox	45.2
vm_get_region_info	11.8	event_notify	46.2
vm_set_region_info	48.1	event_listen	33.2
pm_get_phys_page_set_info	12.7	event_check_ebox	8.7
pm_set_phys_page_set_info	4.4		

表2　スケジューリング・コスト（マイクロ秒）

	ベスト	ワースト
スレッドスイッチ	23.5	42.0
周期→周期（同一グループ）	52.2	74.3
周期→周期（異グループ）	50.5	69.0
周期→通常	36.9	67.8

などのコストが加わる。

アイソクロナス・スケジューラのスケジューリング・コストの測定のため、以下を行なった。

1）通常スレッドから通常スレッドへのスレッド・スイッチコストの測定[注9]

2）周期スレッドから周期スレッド（二つの周期スレッドは異なるスレッド・グループに属する）へのスレッド・スイッチコストの測定

3）周期スレッドから周期スレッド（二つの周期スレッドは同一のスレッド・グループに属する）へのスレッド・スイッチコスト（＝周期スレッド間の同期実現コスト）の測定

4）周期スレッドから通常スレッドへのスレッド・スイッチコストの測定

5）通常スレッドから周期スレッドへのスレッド・スイッチコストの測定

2）、3）、4）及び5）と1）のコストの差が、周期スレッドのスケジューリングのために余分に要するスケジューリング・コストと見積もれる。5）のコストは3）のコストと殆ど差異がないため、1）～4）の測定結果を表2に示す。

この測定結果より、アイソクロナス・スケジューラは、通常のスレッド・スイッチに13～32マイクロ秒の実行コストの付加により実現可能であることがわかる。

3.3　資源競合発生時の周期駆動性能の測定

HiTactixでは、カーネル内資源の排他アクセス制御を細粒度のプリエンプト制御により実現し、資源競合による周期スレッドのスケジューリング遅延時間を小さくしている。カーネル内のプリエンプト禁止区間の最大走行時間が、上記遅延時間の最大値となる。

ユーザ・スレッドからHiTactixの各システムコールを発行し、それぞれのシステムコール処理ルーチンにおけるプリエンプト禁止区間の最大走行時間を測定し、上記遅延時間の最大値を見積もった[注10]。測定結果を表3に示す。

表3から、HiTactixでは、カーネル内の資源競合による周期スレッドのスケジューリング遅延時間は、最大56マイクロ秒程度であることがわかる。

3.4　非同期イベント発生時の周期駆動性能の評価

HiTactixでは、非同期イベント処理をFLIH、SLIH、TLIHの3階層構造を持つ割り込みハンドラによって行

注8 このスレッド・スイッチのコストは、スケジューリングされていたスレッドのユーザ・モードでの実行が停止してから、次にスケジューリングされるべきスレッドがユーザ・モードで実行を開始するまでの時間を示す。他のスケジューリング・コストに関しても同様である。

注10 HiTactixでは、システムコールの引数などの違いなどによりプリエンプト禁止区間の実行時間が大きく変動することがないようにコーディングされている。

表4　FLIH、SLIHの実行時間（／パケット）

FLIH、SLIH 実行時間	60.3マイクロ秒

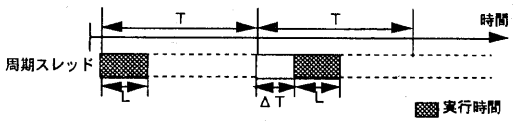

図7　ジッタの定義

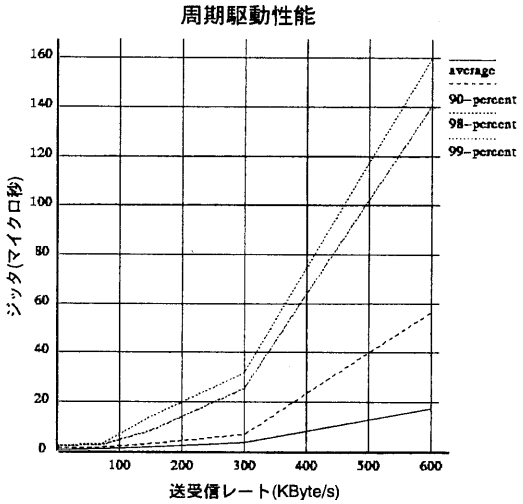

図8　周期と1周期あたりの実行時間の要求

なう。FLIH、SLIHの実行中は周期スレッドの実行が中断するので、FLIH、SLIHの実行時間を極小化することが、周期スレッドの周期駆動性能を保つために重要となる。

通常スレッドが、キャッシュ、TLBを全パージする無限ループを実行中に1パケットが到達した際の、FLIHの起動からTLIHの起動までの時間（周期スレッド実行中に1パケットが到達した際に、FLIH、SLIHの実行により周期スレッドの実行が中断する時間に等しい）を測定した。測定結果を表4に示す。

さらに、T=40ms、L=8msである4つの周期スレッドを周期駆動させつつ、可変レートのパケットの送受信を行なった場合のジッタを変動を測定した。ここで言うジッタとは、図7に示す様に、カーネル内の資源競合や、FLIH、SLIHの実行などにより発生しうる周期スレッドのスケジューリング遅延ΔTのことを指す。

測定結果を図8に示す。測定は、可変レートのパケットの送受信を行ないつつ周期スレッドの周期駆動を10,000回行ない、それぞれのジッタを測定した。得られたジッタの値を昇順にソートし、9,000番目、9,800番目、9,900番目の値をグラフにプロットした（それぞれ、90%、98%、99%の確率で、ジッタがプロットした値より小さく収まることを示す）。全ジッタの平均値も併せてプロットした。この測定結果から、アイソクロナス・スケジューラは、600KByte/sのパケットの送受信を行ないつつ、99%の確率でジッタ160マイクロ秒以内での周期スレッドの周期駆動が可能であることがわかる。

4　まとめ

本稿では、連続メディア処理向きマイクロカーネルHiTactixのリアルタイム・スケジューリング機構の概要として、アイソクロナス・スケジューラ、優先度逆転解消方式、非同期イベント処理方式につき述べた。さらに、アイソクロナス・スケジューラの周期駆動性能評価を行なった。評価の結果、以下を確認した。

1）スレッド・スイッチのコストに13～32マイクロ秒のコスト付加により周期スレッドのスケジューリングが可能である。
2）カーネル内資源競合による周期スレッドのスケジューリング遅延を56マイクロ秒以下に抑えられる。
3）600KByte/sのパケットの送受信をしつつも、99%の確率で周期スレッドのスケジューリング遅延を160マイクロ秒以下に抑えられる。

参考文献

[1] M.Dertouzos, "Control Robotics: The Procedural Control of Physical Processes", Proceedings IF IP Congress, 1974.

[2] Clifford W. Mercer, Stefan Savage, and Hideyuki Tokuda, "Processor Capacity Reserves: Operating System Support for Multimedia Applications", Proceedings of International Conference on Multimedia Computing and Systems, pp90-99, 1994.

[3] Lui Sha, Ragunathan Rajkumar, and John P. Lehoczky, "Priority Inheritance Protocols: An Approach to Real-time Synchronization", IEEE Transactions on Computers, Vol. 39, No. 9, Sep. 1990.

[4] Jay K. Strosnider, John P. Lehoczky, and Lui Sha, "The Deferrable Server Algorithm for Enhanced Aperiodic Responsiveness in Hard Real-Time Environments", IEEE Transactions on Computers, Vol. 44, No. 1, Sep. 1995.

[5] 岩嵜他，"連続メディア向きマイクロカーネルHiTactixの設計と評価", 情報処理学会コンピュータシステムシンポジウム, 1996.

[6] 岩嵜他, "連続メディア処理向きマイクロカーネルの開発（1～5）", 情報処理学会第53回全国大会予稿集, 1996.

[7] 中島達夫, "連続メディア処理に適したオペレーティングシステム", 情報処理学会コンピュータシステムシンポジウム, 1996.