システムソフトウェアと
オペレーティング・システム 92−13
（２００３．２．２８）

組み込み用オペレーティングシステム『開聞』における リアルタイム機能の開発

堀口　努　　並木　美太郎

東京農工大学大学院工学研究科

近年，組込みシステムの多方面への応用に伴って，制御を対象とした従来のハードリアルタイム処理に加えてマルチメディア等のソフトリアルタイム処理を実行させる要求が高まっている．これら 2 種類のリアルタイム処理を同一システムで管理するためには, 従来の組込み用 OS が持つ優先度方式のスケジューリング機能だけでは対応できないという問題がある．そこで，本稿では，従来の優先度が設定された処理に加えて，時間制約として起動周期またはデッドラインが与えられた処理のソフトリアルタイム性を保証すること，デッドラインミス時に処理を実行できることを目標として設計したリアルタイム機能について述べる．

Development of　Realtime Mechanisms for the Embedded Operating System ”KAIMON”

Tsutomu Horiguchi　Mitarou Namiki
Graduate School of Technology, Tokyo University of Agriculture and Technology

In recent years, as embedded comes to be used for many purpose, in additon to perform the hard-realtime processing such as the conventional processing for device control, the soft-realtime processing such as multimedia also comes to be required. To manage these two kind of realtime processing by the same system, it cannot respond only by the priority shceduler mechanism. In this paper, We describe about realltime mechanisms which aims at performing soft-realtime processing given time restlictions such as cycle and deadline of tasks, in addition to performing the conventional hard-realtime processing given priorities.

1　はじめに

近年，組込みシステムの多方面への応用に伴って，制御を対象とした従来のハードリアルタイム処理に加えてマルチメディア等のソフトリアルタイム処理を実行させる要求が高まっている．従来の組込み用 OS が備えるリアルタイム機能は，デッドラインミスが許されないハードリアルタイム処理が要求された．組込みハードリアルタイム性を保証するために特定のデバイスなどの用途に特化して開発された．しかし，近年の組込み用途の多様化に伴い，マルチメディア等のソフトリアルタイム性を保証する要求が高まってきた．このような処理では，デッドライン保証の制限を緩める代わりに特定の用途に特化せず，動画再生を実行しながらネットワーク処理を実行するといった汎用的な利用が可能なことが求められる．これら 2 種類のリアルタイム処理を同一システムで管理するためには，従来の組込み用 OS が持つハードリアルタイム向けの機能だけでは対応できない．そこで，本稿では，従来のハードリアルタイムに加えて，時間制約として起動周期またはデッドラインをもつ処理のソフトリアルタイム性を保証す

ること，デッドラインミス時に処理を実行できることを目標として設計したリアルタイム機能について述べる．

2　組込み用 OS の考察

本章では，代表的な組込み用 OS のリアルタイム機能について考察する．

(1) μITORN

μITRON は組込み用 OS の主流であり，デッドラインの保証を前提とした従来のハードリアルタイム処理向けの仕様となっている．そのため，組込み用途の多様化に伴うマルチメディア処理などに必要なタスク周期やデッドラインなどの時間制約を保証するスケジューラや，タスクの周期実行やデッドラインミス時のハンドリングといったソフトリアルタイム向けの機能が不十分であるという問題がある．

参考文献：[4]

(2) RT-Linux

Linux は近年，組込み用 OS としても注目されている．RT-Linux は，リアルタイムカーネル上で動作するタスクのひとつとして Linux カーネルを動作させることによりハードリアルタイム処理を可能にしている．しかし，Linux カーネル本体にリアルタイム機能を持たないため，Linux カーネル上で動作する動画再生などのソフトリアルタイム性が保証できないという問題がある．

参考文献：[5]

組込み用 OS は従来，デッドラインミスが許されないハードリアルタイム処理向けに開発されてきたため，デッドラインミスが許されるソフトリアルタイム処理向けの機能が不十分であるという問題がある．近年の組込み用途の多様化にともないソフトリアルタイム処理にも対応可能な機能を持つことが組み込み用 OS に求められる．

3　リアルタイム機能の目標

本章では，本機能の目標をまとめる．

(1) 従来のハードリアルタイム性に加えて時間制約が設定された処理のソフトリアルタイム性を保証する．

組込み用 OS が，従来のハードリアルタイム性に加えてタスクやデバイスの起動周期，デッドライン等の時間制約を設定されたアプリケーションのソフトリアルタイム性を保証できるようにする．ソフトリアルタイム処理では，動画再生処理など，周期的な処理を行う一方で，ネットワーク処理など，一定の応答性を求める非周期的な処理を実行することが必要であるため，このような合同スケジューリングを実行できるようにする．本稿では，起動周期が設定されたタスクを周期的タスクと呼ぶ．また，デッドラインが設定されたタスクをデッドライン付きタスク呼ぶことにする．

(2) デッドラインミス時の処理を実行できる

従来のハードリアルタイムに加えて，ソフトリアルタイムに対応するため，デッドラインミスした場合の回復処理を行う機能が重要である．そこで，本機能では，デッドラインミス時の処理を実行できるようにする．本稿では，デッドラインミス時の処理をデッドラインミスハンドラと呼ぶ．

(3) メモリ消費を抑える

近年の，ハードウェアの性能向上に伴い，組込みシステムで利用できるメモリの容量も大容量化している．しかし，組込みシステムでは，様々な性能のハードウェアが用いられるという特徴がある．そこで，メモリ消費をできるだけ抑える設計にする．

以上の事項を目標に，リアルタイム機能を設計した．

4　リアルタイム機能の全体構成と特徴

4.1　リアルタイム機能の全体構成

リアルタイム機能の全体構成を図 4.1 に示す．

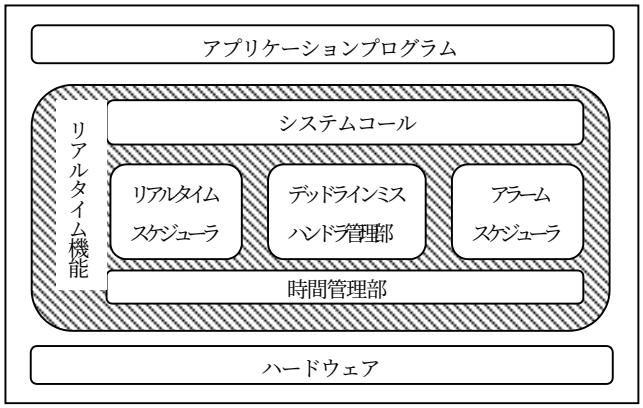

図 4.1　リアルタイム機能の全体構成

リアルタイム機能は次の資源管理部から構成される．

(1) リアルタイムスケジューラ

周期やデッドラインが設定されたタスクのリアルタイム性を保証するように実行順序を決定する．

(2) デッドラインミスハンドラ管理部

デッドラインミス時の処理を管理する．

(3) アラームスケジューラ

デバイスの起動時刻のハードリアルタイム性を保証する機能を持つ．

(4) 時間管理部

デッドラインミスハンドラや，タスクやデバイスを指定時刻に起動のするために必要なシステム時間の初期化，更新を行う．

(5) システムコール部

アプリケーションはシステムコールで本機能を使用する．

4.2 リアルタイム機能の特徴

本リアルタイム機能は次の特徴を持つ．

(1) 複数のアルゴリズムを有するリアルタイムスケジューラ

多様なリアルタイム処理に対応するために複数の RT スケジューリングアルゴリズムを持つという特徴がある．タスクに設定された，優先度に加えて周期やデッドライン等の時間制約に応じて，優先度方式，RM 方式，EDF 方式やバックグラウンド方式でスケジューリングを行えるようにした．

(2) アプリケーションが登録可能なデッドラインミスハンドラ

ソフトリアルタイムにともなうデッドラインミスに対応するために，アプリケーションが周期やデッドラインを設定し，関数形式で記述したデッドラインミスハンドラをデッドラインミス時に実行できるようにした．デッドラインミス時に即座に回復しなければならない要求に対応するために，優先度をハンドラ登録時にアプリケーションが選択可能にした．

(3) デバイスの起動時刻を保証するアラームスケジューラ

従来の組込み用途であるデバイスの起動時間のハードリアルタイム性を保証した処理を実行可能なアラームスケジューラを持つ．

本機能は以上のような特徴をもつ．

5 リアルタイム機能の設計

本章では，本機能各部の設計について述べる．

5.1 リアルタイムスケジューラ

本機能のリアルタイムスケジューラは，多様化したリアルタイム処理に対応するために，複数のアルゴリズムを有する特徴を持つ．本機能が持つリアルタイムスケジューラを次に示す．

(1) RM スケジューラ

周期的タスクを，周期が短い順に優先度を与えるレートモノトニック（RM）方式でスケジューリングする．

(2) EDF スケジューラ

デッドライン付きタスクを，絶対デッドラインが早い順に優先度を与える EDF 方式でスケジューリングする．

(3) デッドライン(DL)スケジューラ

デッドラインミス時に即時実行がもとめられるデッドラインミスハンドラを，デッドラインスケジューラがスケジューリングする．

(4) 優先度スケジューラ

優先度が与えられたタスクをスケジューリングする．

(4) メタスケジューラ

(1)から(4)のスケジューラを呼び出す役割を持つ．

本機能が持つリアルタイムスケジューラの構成を下図にまとめる．

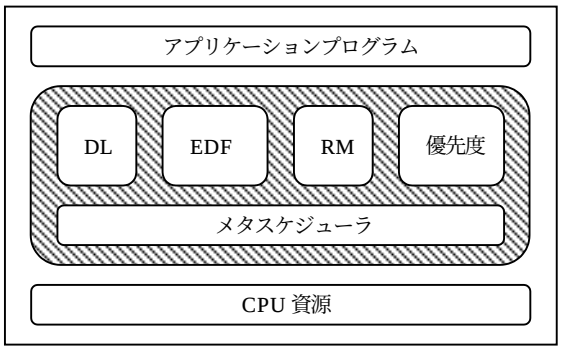

図 5.1 リアルタイムスケジューラの構成

各リアルタイムスケジューラは，メタスケジューラによって呼び出される．スケジューラはスケジューリング完了したかを呼出し側に返す外部仕様に統一している．このため，メタスケジューラがスケジューラを呼び出す順序を変更するだけで，タスクの優先度を容易に変更できるという特徴がある．

5.1.1 EDF スケジューラ

EDF スケジューラは，ソフトリアルタイム処理における，ネットワーク処理など非周期に発生し，デッドラインまでに応答性を求める処理のための機能である．タスクに与えられたデッドラインが早い順にスケジューリングする．タスクのデッドラインを次のシステムコールで設定できるようにした．

表 5.1　デッドライン設定システムコール

名称	引数	リターン値	機能
SetDeadline	デッドライン(ms)	なし	タスクのデッドラインを設定する

このシステムコールの特徴は，デッドラインの設定と解除を同一の形式で実行することができるということである．このことによってデッドライン設定とクリア処理を単純化している．デッドラインを設定したい場合は，引数に相対デッドラインを指定する．デッドラインを保証後に，デッドラインミスが生じないように，デッドラインを解除する必要がある．この場合，引数に 0 を設定する． EDF スケジューラの設計を下図にまとめる．

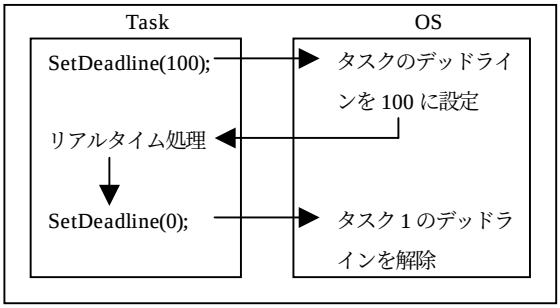

図 5.1　EDF スケジューラの設計

図 5.1 は，ソフトリアルタイム処理へデッドラインを設定している．リアルタイム処理に，SetDeadline(100) ;で 100ms 後にデッドラインを設定する．そして，リアルタイム処理が終了したら SetDeadline(0);でデッドラインを解除する．このように同一のシステムコールで，ソフトリアルタイム処理へデッドラインの設定と解除をできるようにした．

5.1.2　RM スケジューラ

RM スケジューラは，ソフトリアルタイム処理の動画再生などの周期的タスクを実行するための機能である．周期が短い順に実行する RM 方式でスケジューリングする．本機能の特徴は，次の 2 つのシステムコールで周期的タスクを実行することである．

表 5.2　周期設定システムコール

名称	引数	リターン値	機能
SeCycle	初期位相 周期	なし	タスクの周期を設定する
EndCycle	継続（1） 終了（0）	なし	1 周期終了を OS に通知する

SetCycle は，起動周期だけではなく初期位相も指定できる特徴を持つ．初期位相をずらして設定することによって，同時刻の起動による周期的タスクが待ち状態になるのを防ぐことが出来る．プログラムを容易にするために, 初期位相は相対時間で設定し, OS が絶対時間に変換するようにした．周期的タスクは 1 周期が終了するごとに，次の起動時刻とデッドラインを OS に再設定させる必要がある．このため，1 周期終了したら，EndCycle で通知するようにした．EndCycle の特徴は，周期的起動の継続と終了を選択可能なことである．このように，本機能では, SetCycle と EndCycle を用いて, 周期を設定し, RM 方式でスケジューリングする．RM スケジューラの設計を下図にまとめる．

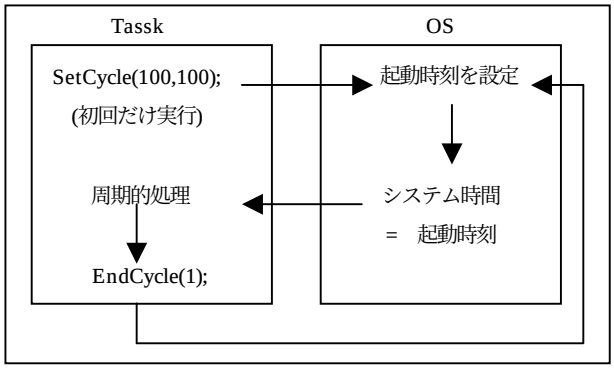

図 5.2　RM スケジューラの設計

周期を SetCycle で指定すると, 周期的起動が開始される．1 周期処理後に EndCycle を実行することによって，周期的起動を継続または終了を選択できる．以上のように，本機能は 2 つのシステムコールで，周期的タスクを実行するという特徴を持つ．

5.1.3　デッドラインスケジューラ

デッドラインスケジューラは，ソフトリアルタイム処理に伴うデッドラインミス時の処理を即時実行するための機能である．ミスハンドラをミス時に即時実行する場合，他のタスクのデッドラインミスを誘発する可能性がある．そこで，ユーザがミスハンドラの即時実行を選択が可能な特徴を持たせた．本スケジューラはユーザが即時実行を選択したハンドラがスケジューリング対象である．デッドラインミスハンドラの即時実行選択機能の詳細は, 5.2 節で述べる．

5.1.4　メタスケジューラ

ソフトリアルタイム処理では，動画再生などのタスクの周期性を保証する一方でネットワークなどの

非周期的なタスクのデッドラインを保証する処理が求められる．本機能では，様々な時間制約が与えられた処理を次に示す優先度で実行するようにした．

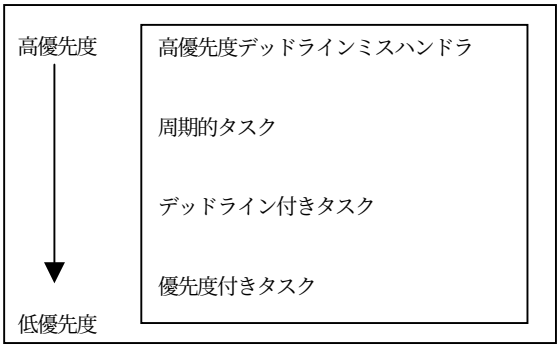

図 5.3　タスクの優先順位

合同スケジューリングは様々な方式があるが，本機能は組込み向けなので，メモリ消費やオーバヘッドが少ないバックグラウンド方式でスケジューリングするようにした．バックグラウンド方式は，低優先度の処理が高優先度のバックグラウンドで実行する．本機能では，デッドラインミス時に即時実行させたいハンドラを最優先で実行することでミス時に即時実行できるようにした．また，タスクの周期性の保証を優先し，非周期的なデッドライン付きタスクものは，周期的タスクのバックグラウンドで実行するようにした．この方式を実現するために，次図に示す順序でスケジューラを実行する．

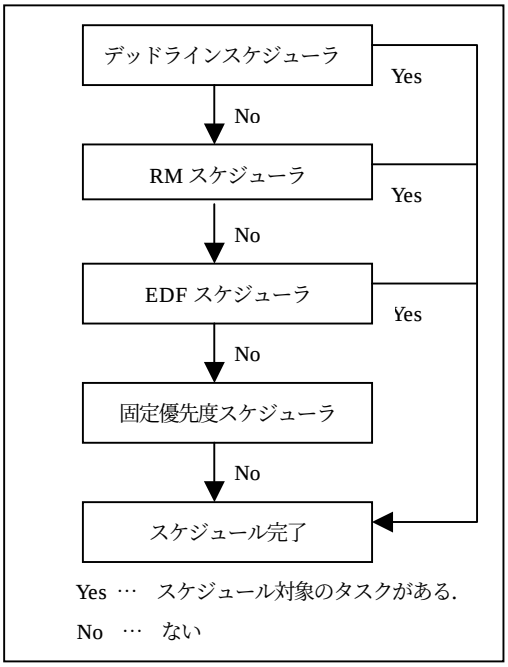

図 5.4　メタスケジューラの処理

以上のように，メタスケジューラは，バックグラウンド方式で合同スケジューリングを行う．

5.2　デッドラインミスハンドラ管理部

デッドラインミスハンドラ管理部は，デッドラインミス時のハンドラを管理する役割を持つ．デッドラインミスハンドラ管理部は次のような特徴を持つ．

(1) 関数形式で記述したデッドラインミスハンドラをアプリケーションが登録できる．
(2) デッドラインミスハンドラをデッドラインミス時に即時実行するかをユーザが選択できる．
(3) デッドラインミスハンドラをミスしたタスクと同一のコンテキストを使用することでメモリ消費を抑える．

次の 2 つのシステムコールでデッドラインミスハンドラを実行する．

表 5.3　ミスハンドラのシステムコール

名称	引数	リターン値	機能
SetDlmisHandle	エントリポインタ 優先度フラグ	なし	デッドラインミスハンドラを登録する
EndDlmisHandle	エントリポインタ	なし	ミスハンドリングの終了を OS に通知する

ハンドラをミス時に実行するため SetDlmisHandle でデッドライン設定前に登録する．ハンドラは，原則的にミスタスクの優先度に従うが，ミス時にハンドラを即時に実行する要求に対応するために，登録時に優先度フラグで選択可能にした．フラグがオンの高優先度ハンドラは，デッドラインスケジューラでミス時に即時実行される．デッドラインミス時の OS の処理を図 5.4 にまとめる．

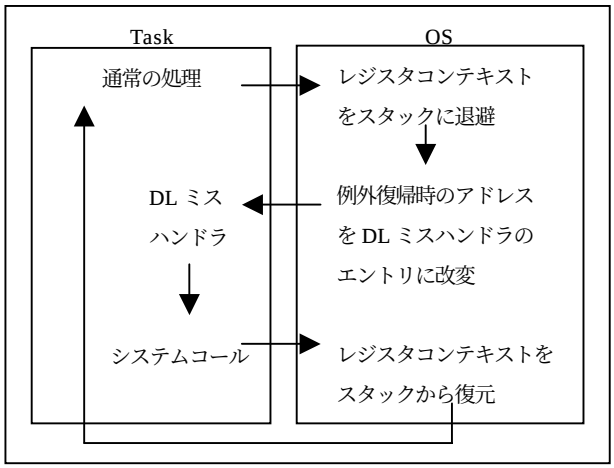

図 5.5　デッドラインミスハンドラの設計

デッドラインミスが発生すると，OS は，次の処理を行う．

(i)ミスタスクのコンテキストをスタックに格納する．

(ii) 例外復帰時のアドレスをデッドラインミスハンドラのエントリポインタに設定する.

その後，ミスタスクがディスパッチされると，ミス時の処理を中断し，ミスハンドラが実行される．終了後， OS にミスハンドリングが終了したことを通知するために EndDlmisHandle を実行する. そして，OS は次の処理を行う.

(iii) スタックからタスクのレジスタコンテキストを復元する.

以上のように本機能は，タスクのコンテキストを使用して，タスクが登録したデッドラインミスハンドラを実行するという特徴を持つ.

5.3 時間管理機能

本機能は，周期起動やミスハンドラを起動する基準となるシステム時間を管理ための機能である．本管理機能の特徴は，組込みシステムの多様な性能要求に対応するために，時間更新間隔をユーザが変更可能なことである．時間管理機能では次に示す処理を行う.

(1) システム時間の初期化

システム時間をリアルタイム処理開始前に初期化する.

(2) システム時間の保持，更新

システム時間開始時からの絶対時間を保持する．また，一定の間隔でシステム時間を増加させる.

(3) システム時間の更新間隔の変更

システム時間の更新間隔は，短いほどシステム時間の精度を向上させることができるが，オーバヘッドが増加する．逆に更新間隔を長くすると，オーバヘッドは減少するが，時間の制度が低下する．そこで, その選択をユーザプログラムが出来るようした. 初期設定では，1ms 間隔で更新するが，その間隔をクロック単位で変更できるようにした.

これらの機能を提供するシステムコールを次にまとめる.

表 5.5　時間関連システムコール

名称	引数	リターン値	機能
GetSystime	なし	システム時間	システム時間の取得
ChangeTImerInterval	クロック数	なし	時間更新間隔の変更

以上のようにして，時間管理機能は，システム時間の保持，更新を行う．本機能は，組込みの多様な性能要求に対応するために，時間更新間隔をユーザが変更可能という特徴を持たせた.

6　アラームスケジューラ

アラームスケジューラは，従来の主要な組込み用途であるデバイスの起動時刻を保証するための機能である．アラームスケジューラの特徴は，デバイス制御に必要な起床時間や周期等の情報を保持する ACB にデバイスを仮想化する事である．ACB はタスクコントロールブロックより百数十バイト小さいので，メモリ消費を抑えることができる.

```
struct {
        typeList list;      /* 双方向リスト */
        long time;          /* 次回起床時間 */
        long type           /* 周期間隔 */
        void (*entry)();    /* 起床させるドライバ */
        int priority;       /* 優先度 */
        int state;          /* スケジュールされてるか */
}typeAlarm;
```

図 6.1　アラームコントロールブロック

ACB はアラームリストというリストに起動時間が早い順に接続し，ACB を起動時刻に起動する.

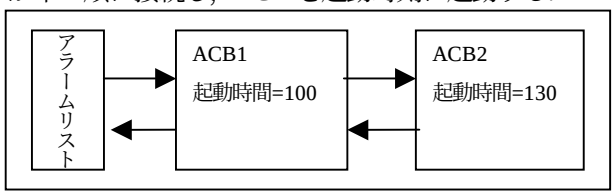

図 6.2　アラームリスト

ユーザが，次のシステムコールで ACB を追加，削除を行えるようにした.

表 6.1 アラームスケジューラのシステムコール

名称	引数	リターン値	機能
SetAlarm	ACB の起床時間 ACB へのポインタ	なし	ACB をアラームスケジューラに登録する.
DelAlarm	ACB へのポインタ	なし	ACB をアラームスケジューラから削除する.

(1) ACB をアラームスケジューラに登録する.

ACB の特徴は, ユーザ領域に持たせることにより，アプリケーションが登録可能なことである．デバイスドライバや時間制約などの情報を記述した ACB を作成し，SetAlarm システムコールで登録する．アラームスケジューラは, ACB が保持する起動時刻になったら，アラームコントロールブロックに登録されたデバイスドライバを実行する.

(2) アラームコントロールブロックをアラームスケ

ジューラから削除する．

不必要になった ACB は，DelAlarm システムコールで削除する．

次にアラームスケジューラの設計をまとめる．

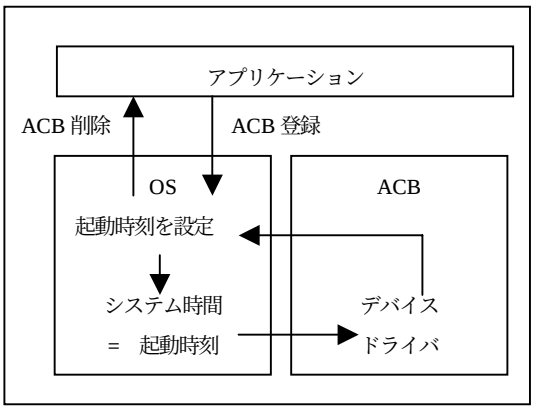

図 6.3　アラームスケジューラの設計

以上でアラームスケジューラの設計を示した．アラームスケジューラでは，デバイスを ACB に仮想化するという特徴がある．そして，ユーザが ACB に設定した起動時刻を保証する役割持つ．

7　リアルタイム機能の実装と評価

本リアルタイム機能を実装したボードの仕様を下表に示す．

表 7.1 開発環境

開発言語	C 言語
コンパイラ	exeGCC
CPU	VR4300
CPU 動作周波数	96MHz
メモリ	4M バイト

C 言語を用いて組込み用 OS『開聞』に実装した．MIPSVR4300 が搭載されている組込み用ボードを用いた．動作周波数は 96MHz であり，4Mbyte の RAM が搭載されている．この環境でシステムコールのオーバヘッドを計測した結果表 7.2 のようになった．最も実行時間が短いのは，SetDlhandle であった．これは，コンテキストのエントリポインタをデッドラインミスハンドラに改変するという，命令数が少ない処理を行っているからである．一方最も実行時間がかかるのは，EndDlmisHandle であった．これは，デッドラインミスしたタスクのコンテキストをスタックから復元するという命令数が多い処理を行っているためである．

表 7.2 テムコールのオーバヘッド

名称	μs	機能の概要
SetCycle	31.5	周期を設定する
EndCycle	10	1 周期終了を通知する
SetDlhandle	3.37	DL ミスハンドラを登録する
SetDeadline	6.83	デッドラインを設定，クリアする
EndDlmisHandle	38.16	DL ミス処理の終了を通知する
ChangeTimerInterval	8.2	システム時間更新間隔を変更する
Get_Systime	4.6	システム時間を取得する
SetAlarm	5.3	ACB を登録する
DelAlarm	4.9	ACB を削除する

時間制約を保証する目標を検証するために次の処理で構成される，パケット送受信プログラムを作成した．

(a) LAN カード用デバイスドライバ

パケットを受信し，送信パケットを作成する．

(b) 負荷タスク

1s 周期で起動しボードに負荷をかける．

(c) パケット送信タスク

パケットを送信する．

パケット送信タスクを周期的タスクとして実行した場合と実行しない場合について比較評価をおこなった．

(1) 周期的タスクとして実行しない場合

パケット送信タスクを時間制約を持たない優先度付きタスクで実行した場合の実行結果を次図に示す．

```
Pinging 192.168.0.2 with 32 bytes of data:
Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 514ms, Maximum =  551ms, Average =  535ms
```

図 7.1　本機能を用いない場合の応答時間

この図から最悪応答時間に 551ms かかっていることがわかる．パケット送信タスクが負荷タスクのバックグラウンドで実行されるため応答性能が悪くなる．

(2) 周期的タスクとして実行した場合

次に，パケット送信タスクを周期 1ms の周期的タスクで実行した場合の実行結果を図 7.2 に示す．

この場合，パケット送信タスクが RM スケジューラでスケジューリングされる．このため，最悪応答時間は，6ms と良くなっている．パケット送信タスクを周期的タスクとして扱うことによって，扱わない場合の約 90 分の 1 に短縮することが出来た．

```
Pinging 192.168.0.2 with 32 bytes of data:
Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 5ms, Maximum =  6ms, Average =  5ms
```

図 7.2　本機能を用いた場合の応答時間

以上の結果より，時間制約が与えられた処理のリアルタイム性を保証するという目標を達成することができた.

8　おわりに

本稿では，組込み用 OS『開聞』において開発したリアルタイム機能について述べた．ここでは，研究を通じて得られた成果について述べる.

(1) 時間制約が設定された処理のソフトリアルタイム性を保証できるようにした.

組込み用途の多様化に伴うソフトリアルタイム処理に対応するために，従来の優先度に加えて起動周期, デッドラインをタスクに設定できるようにした.

本機能には，与えられた時間制約を保証するために，RM 方式，EDF 方式，バックグラウンド方式など複数のアルゴリズムを有する特徴を持つ．本機能によって，組込み用 OS を従来のハードリアルタイムに加えてにソフトリアルタイムにも対応できるようにした.

(2) 周期的タスクとデッドライン付きタスクが同時に存在し，実行できるようにした.

多様化したリアルタイム処理では，動画再生処理などの，周期的タスクを実行する一方で，ネットワーク処理など，非周期的タスクのデッドラインを保証する合同スケジューラが必要である．本機能では，組込みシステムの多様なメモリ資源に対応するために，メモリ消費が少ないバックグラウンド方式による合同スケジューリングを実行できるようにした.

(3) デッドラインミス時の処理を実行可能にした.

ソフトリアルタイム処理では，デッドラインミス時の処理が重要である．そこで，アプリケーションの関数形式でハンドラを登録し，ミス時に実行できる ようにした．ミスハンドラはミスしたタスクのコンテキストを使用することによって，メモリ消費を抑えた．ミスハンドラの優先度をハンドラ登録時にユーザが選択することで，ミス時に即座にハンドリン可能な特徴を持たせた.

(4) デバイスのスケジューリングを可能にした.

従来の主要な組込み用途であるデバイスのリアルタイム性を保証するために，アラームスケジューリング機能を実現した．アラームスケジューラでは，デバイスのドライバや時間制約などの情報を ACB に仮想化する特徴を持つ．アラームスケジューラは，ACB を指定時刻に起動する役割を持つ．アラームスケジューラによって，従来の主要な組込み用途であるデバイスのリアルタイム性を保証できるようにした.

今後の課題として,『開聞』のアプリケーション拡充による本機能の検証が挙げられる．本研究で検証したネットワーク処理に加えて，動画再生等のソフトリアルタイム性が要求されるアプリケーションを拡充し，本機能を検証する必要がある.

参考文献

[1] 中本　他, 組込みシステム技術の現状と動向, 情報処理 Vol38 No.10, 1996.

[2] 藤波　他, Muse オペレーティングシステムにおけるリアルタイム機能について, 情報処理学会研究報告「オペレーティングシステム」No48-3, pp.17 - pp.24, Sep.1990.

[3] 市瀬　他, UNIX におけるリアルタイム性導入に関する一考察, 情報処理学会研究報告「マイクロコンピュータとワークステーション」No54 - 4, pp.1 - pp.7, Feb.1989.

[4] μITRON 4.0 仕様 Ver.4.00.00,TRON 協会

[5] LINUX　リアルタイム計測/制御開発ガイドブック, 秀和システム

[6] Gafford,j. D. ,Rate Monotonic Scheduling, IEEE Micro,pp.34-38,1991.

[7] Supri,M.: Analysis of Deadline Scheduled Real-Time Systems, Technical Report No.2772,INRIA,1996.

[8] Stankovic, J. A., Spuri, M., Ramamritham, K. and Buttazzo, G. C.: Deadline Scheduling for Real-Time Systems: EDF and Related Algorithms, Kluwer Academic Publishers, 1998.

[9] Lehoczky, J. P. and Ramos-Thuel, S.: An Optimal Algorithm for Scheduling Soft-Aperiodic Tasks in Fixed-Priority Preemptive Systems, in Proceedings of the IEEE Sympsium on Real-Time Systems, pp. 110-123, 1992.