4K−5 リアルタイムOSにおける高速応答性能の実現と透過的なOSモデルの実現

市瀬 規善，大橋 範夫，下島 健彦，古城 隆

日本電気(株)C&C共通ソフトウェア開(本)基本システム開発部

1 はじめに

近年，リアルタイムシステムにおいては，組み込み用OSとしての性質上，そのサイズや処理速度に加えて，リアルタイム性能上もっとも重要なものの一つである割り込み応答性能が要求されている．ここで，特にリアルタイムシステムにおける割り込み応答性能は，アプリケーションによるところが大きい処理速度と違い，OSの性能で決まることが多いので，近年OSに対する要求が高まっている．

また，リアルタイムシステムを構築する際，開発体制の関係から，OSは，ある種のユーザに対してはOSの動作に対する透過性を持たせて，システムを最適化できるようなモデルを提供し，他のユーザには，ハードウェアを完全に隠蔽したモデルを提供する必要がある．

今回開発した，リアルタイムOS rtos V3.0 は，ITRON Iに準拠し，リアルタイムシステムに必要な機能を提供する．また，割り込み遅延時間については，(ソフトウェアとして)0という性能を実現したOSである．

また，本リアルタイムOSは，ユーザからみたOSの透過性を高くするため，ユーザに対するインタフェースをシステム(OS回り)構築者用，および，その他ユーザ(アプリケーション構築者)用のインタフェースを用意した．システム構築者用インタフェースでは，OS資源の取り扱いやプロセッサリソースなどをユーザに開放し，動作などの透過性を高くし，制約条件を守ったシステム設計を容易にしている．アプリケーション構築者用のインタフェースでは，ITRON準拠のモデルを提供している．

2 性能

2.1 割り込み遅延時間

割り込み応答性能には次のようなものがある（図1）．

割り込み禁止時間:
　割り込みが入ったにもかかわらず，割り込みが禁止されているために受け付けられない間の時間

割り込み遅延時間:
　上記の割り込み禁止時間に加えて，割り込みが受け付けられて，割り込み処理が動き出してから，OSによる割り込み処理の前処理を経，ユーザが登録した割り込みハンドラの実行が開始されるまでの時間

OSによる割り込みのオーバヘッド:
　割り込み処理が実行される際にOSにより消費される時間（割り込み処理の前処理，後処理）

本リアルタイムOSでは，OS内での割り込み禁止区間をなくすことで（CISCチップ上で），割り込み禁止時間をソフトウェアとして0にしている（CPUの命令の長さ分の割り込み禁止時間は存在する）．

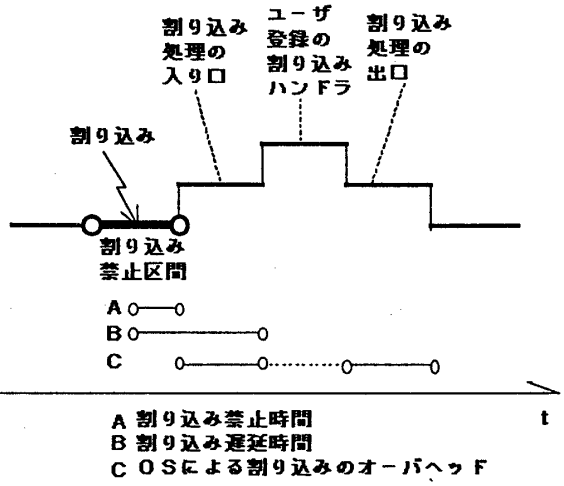

図1 割り込み応答時間

また，割り込み処理の前処理は存在しない．よって，割り込み遅延時間もソフトウェアとして0にしている．

2.2 割り込み禁止時間 0 の実現

タスクや割り込みハンドラ中では，システムコールによってOS内のデータをアクセスしている．そのためデータの正当性を保つためにデータアクセスの排他を行なわなければならない．このために，従来は，割り込みハンドラから発行できるシステムコールがアクセスするデータについては，そのアクセス中は，割り込みを禁止しなければならなかった．

本OSでは，割り込みを禁止しないために，図2のように割り込み処理中のシステムコールは実際の処理は行なわず，遅延して処理するようキューイングしておき，割り込み処理レベルから，タスクレベルへ復帰した後，OSの排他区間を抜けてから行なう様にする．（ただし，この方法の実現のためにはCPUの命令としてcompare and swap 命令を必要とする．）[1]

しかし，この方法では返り値を期待するシステムコールなどは，対応できない．ところが，システム構築上実際に割り込みから発行できる必要があるシステムコールはかぎられている．そこで，割り込み処理から発行できるシステムコールを，リアルタイムシステム構築上必要なもののみに，制限した．割り込み処理から発行可能なシステムコールは，メッセージセンドなどタスク間通信機能を持つシステムコールと，メモリを獲得するシステムコール，タスクに非同期通知をするシステムコールである．タスクの起床処理を遅延しても，タスクが起床されるタイミングが遅れることはない．なぜなら，もしタスクを割り込み処理中で起床したとしても，実際にタスクが起床して動き出すのは割り込

---

implementation of interrupt responce performance on real-time OS

Noriyoshi ICHINOSE, Norio OHASHI, Takehiko SHIMOJIMA
Takashi KOJO
NEC Corporation

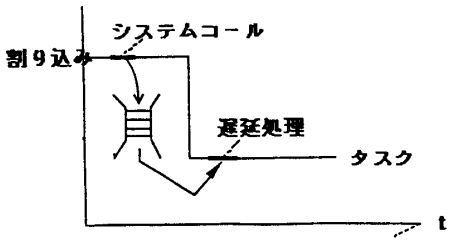

図2 遅延処理

み処理を抜けて，しかもOSの排他区間を抜けた後であり，上記の動作と一致する．

3 構造

3．1 構成

本リアルタイムOSは，その機能の違いから，コアカーネル，ITRON I 準拠カーネル，仮想版カーネルの三種にわけられるが，それらは包含関係になっており，三層構造を呈している．一番下層のコアカーネルは，リアルタイムシステムに必要な機能をコンパクトにまとめたもので，そのインタフェースはITRON I のサブセットとなっている．これは，タスク管理，タスク間通信管理，メモリバッファプール管理、タイマ管理を備えている．二層目のITRON I 準拠カーネルは，カーネルの動的資源管理を強化し，また,割り込み/例外管理をITRON I に準拠させたものとなっている．三層目の仮想版カーネルは，多重仮想空間管理を備えている．(図3)

本OSにおける，テキストおよび初期値ありデータサイズは，コアカーネルで，3.8kbyte，ITRON I 準拠カーネルで7.3kbyteである（V60/70/80版）．

本OSは，ライン数で75%がC言語で記述されているので，移植性が高い．現在，V60/70/80，VR3000版を開発中である．

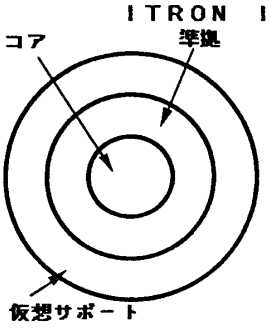

図3 カーネルの構造

3．2 透過性

3．2．1 システム（OS）への機能追加の容易性

本リアルタイムOSは，プロセッサリソースをほとんどユーザに開放している．例えば，V60/70/80版のOSでは，特権レジスタはOSのスタックポインタ用のレジスタ（ISP）を、特殊なケースで使用する以外，使用しない．

これはつまり，本リアルタイムOSは，CPUのステータスなどによらず走行可能であり，ユーザがCPUの機能を利用して実現する場合でも，OSとは独立したモジュールとして，OSの機能に影響を与えることなく構築可能であることを意味している．たとえば，仮想メモリ機能などをOSと独立したモジュールとして簡単に構築できる．

また，ハードウェア依存部を切り離すだけで，OS（システム全体）を簡単にプロセス化できるなど，他OS上でのシュミレーションも簡単にできる．

3．2．2 OS資源と消費メモリ

OSの資源は，ITRON I 仕様で規定されているタスクやメールボックスといった，オブジェクトの一つ一つに対応した資源の他に，オブジェクト管理に必要な間接的な資源が存在する．しかもそれは，OSが暗黙の内に獲得することが多い．間接的資源が多い場合，設定した資源の数に対するメモリ消費量はわかりにくい．

本OSでは，OS資源の設定方法は，間接的資源を暗黙には獲得せず，ユーザが意識できるインタフェースにしている．具体的には，OS資源は適当なメモリ空間をユーザが固定長メモリプールとして初期化し，その間接的資源もユーザが用意する．これをカーネル資源として渡すことにより設定される．

5 まとめ

今回開発したrtos V3.0について、その処理速度は，以下の様になった．なお，ここでは，タスクスイッチと，タスク間通信の内，メールボックスのセンド，レシーブの処理時間を記載する（V70 25MHz キャッシュオフ）．（ただし，ここでいうタスクスイッチは，最高優先度タスクの検索とコンテキストスイッチである）

タスクスイッチ	:30.4
snd_msg	:33.6
rcv_msg	:38.1

（単位:マイクロ秒）

CPUの高機能，高性能化とリアルタイムシステムの多様化にともない，リアルタイムOSに対する要求は，機能，性能はもとより，さまざまなものが要求されている．

今後は，リアルタイム性能として重要視されているものの一つであるプリエンプション時間の保証や，その他ハードリアルタイム指向の機能，OS外付けの部品の充実によるユーザインタフェースの改善および付加機能の充実を検討していきたい．

[1] Henry Massalin and Calton Pu. Threads and input/output int the synthesis kernel. the 12th ACM Symposium on Operating System Principles, 23(5):191-201, December 1989