

μ C T R O N の 試 作

2N-2

大 利 直 行¹⁾, 曾 根 俊 彦¹⁾, 菅 野 淳²⁾
 松 下 通 信 工 業 (株)¹⁾ 松 下 電 器 産 業 (株)²⁾

1. はじめに

CTRON (Central and communication TRON) は、情報処理、通信処理、および交換処理、ワークステーション向けに、リアルタイム指向のオペレーティングシステムのインタフェースを共通化、標準化したものである。このCTRONのカーネルは共通部と選択部に別れ、さらに小規模機器向けに、共通部から最低限必要とされるシステムコールに絞った、μCサブセットが規定されている¹⁾。筆者らは、情報通信分野の小規模機器向けのOSとして、このμCサブセット機能を有するカーネル(μCTRONと呼ぶ)を試作した。

2. プログラムの構成

μCTRONのプログラム構成を図1に示す。図に示されているようにμCTRONは、システムコール単位にプログラムを分割して構成している。ユーザタスクとμCTRONの間は、システムコール割り込みにより接続される。すなわちユーザタスクがμCTRONのシステムコールを発行する際に、割り込み命令(INT, TRAP命令)を使用しカーネルに制御を渡す。カーネルに対し発行されたシステムコールは、カーネル内でその要求されている処理内容により各プログラムをコールし、個別の処理を行う。

3. カーネルの仕様

μCTRONのカーネル本体は、流通性を考慮してC言語で記述し、ハードウェア依存部のみアセンブラで記述してある。その比率はC言語部分が約95%を占める。対象CPUは、当初インテルの8086シリーズをター

ゲットとして開発し、現在モトローラ系の68000シリーズについても、一部のシステムコールについて動作を確認している。CPUによる処理の違いは、主にアセンブラ部分で吸収し、C言語内については、”#if def文”を利用してソースプログラムを共通化している。カーネルの詳しい仕様を表1に示す。

表1. カーネルの仕様

項目	仕様
対象CPU	i8086, i80186, i80286 (一部MC68000で動作)
システムコール数	44個 (CTRON : μCサブセット対応)
タスク数	最大65,535個
タスク優先度	実行レベル (1~256) 優先度 (1~256)
スケジューリング	実行レベル及び優先度 (+タイムスライス)
記述言語	C言語 + アセンブラ

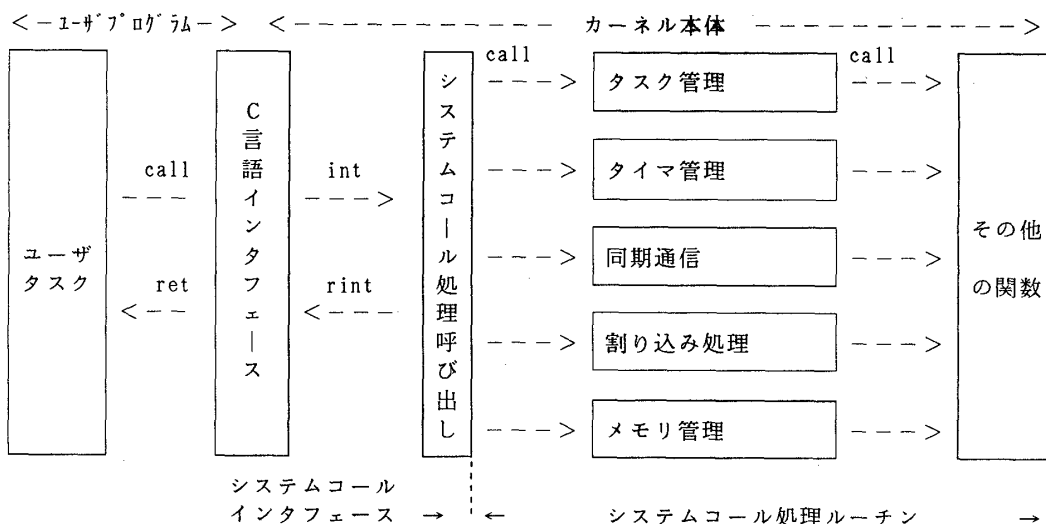


図1. カーネルプログラムの構成

A development of micro-CTRON
 Naoyuki OTOSHI¹⁾, Toshihiko SONE¹⁾, Jun SUGANO²⁾
 Matsushita Communication Industrial Co., Ltd.¹⁾
 Matsushita Electric Industrial Co., Ltd.²⁾

表2. カーネル（試作版）の測定結果

		μCTRON		IGEM	
CPU		80286 10MHz	68000 10MHz	80286 10MHz	68000 10MHz
容量		41Kバイト	37Kバイト	34Kバイト	32Kバイト
処 理 速 度	CRE_TSK DEL_TSK μsec	1050	1670	710	1090
	SLP_TSK WUP_TSK μsec	790	1130	520	780
度	SET_EVF WAI_EVF μsec	1000	1280	810	1110

4. カーネルの評価方法

リアルタイムOSの評価方法として、有効なベンチマークテストはまだ提案されていない。しかし、現在性能評価方法として使用されている指標として次のようなものがある²⁾。

- ① コンテキストSW時間
- ② タスクSW時間
- ③ 最大割り込み禁止時間
- ④ 割り込み応答時間
- ⑤ クロック処理時間
- ⑥ システムコール処理時間

これらの指標のうち、今回の性能評価にあたっては、実際の使用に近い形での評価という観点から、②のタスクSWと⑤のクロック処理時間および⑥のシステムコール処理時間を含めた形式で測定した。具体的な方法としては、複数回のシステムコールを発行し、この前後でのシステムクロックを計測することにより1回当たりのシステムコール処理時間を測定している。

この評価手順を次に示す。

- (1) 優先度が異なる2個のタスクを用意し、システムクロックを0に設定する。
- (2) 優先度の高いタスクは待ち状態に入り、優先度の低いタスクへ制御が渡る。
- (3) 優先度の低いタスクは、優先度の高いタスクを起床させる。
- (4) 上記(2)(3)の処理をN回繰り返し、最後にシステムクロックの値を読み取る。

これにより得られたクロック値は、N回の(システムコール処理時間+クロック処理時間+タスクSW時間)となり、1回当たりの処理時間が測定できることになる。

5. 測定結果

今回測定するにあたり、システムコール時間は2つのシステムコールの組合せで測定した。またあわせて、既に当社で開発しているIGEM (Industrial Global

Environment Manager)³⁾と比較した。表2にその結果を示す。IGEMはITRON (Industrial TRON)仕様のリアルタイムOSであり、ITRONとμCTRONは類似した機能を持っている。

μCTRONはITRONに比べ、エラー情報がかなり詳細に規定されており、エラーチェック処理の負荷がIGEMよりも重いものになっている。また入力パラメータ数もITRONに比べて多く、パラメータをITRONではレジスタ渡しで行っているが、μCTRONではパラメータブロックの先頭アドレスで渡し、実際の処理時に変数にセットしている。このため、μCTRONの処理速度はIGEMよりも2~3割遅いという結果となった。

6. おわりに

以上、本稿ではCTRONカーネルのμCサブセット機能を持つ、μCTRONについて述べた。μCTRONは現在、試作段階にあり、改良の余地を多分に残している。その一つとして、現在アドレスで渡しているパラメータを、図1のシステムコール処理ルーチン入口でシステムスタックに積み上げて渡すことにより、処理速度の短縮化が期待できる。その他、今後のアルゴリズムのチューンナップにより処理速度の高速化を図る予定である。また併せて、今後CTRON仕様に基いた、基本OS機能の拡充も行う予定である。

最後に、松下通信工業、並びに松下電器産業の関係各位に感謝します。

【参考文献】

- 1) 小菊 一三、大久保 利一、松下 政好：「CTRONカーネルインタフェースの拡充」第2回トロン技術研究会 PP.69~86 (1988.10.6)
- 2) 門田 浩、岩崎 保：「ITRONの評価に関する一提案」電子情報通信学会 第3回リアルタイムアーキテクチャTRON研究会 (1987.10)
- 3) 桑鶴 敬一郎、菅野 淳、上田 謙一：「多目的リアルタイムOSの試作」情報処理学会 第35回(昭和62年後期)全国大会論文集