

V70/V80 CTRON カーネルの実現

5P-8

内田 昭宏*, 中本 幸一*, 佐藤 直樹*, 大鷹正之**, 小泉 正*

*日本電気(株) C&C 共通ソフトウェア開発本部

**日本電気(株) 交換第一ネットワークシステム事業部

1 はじめに

CTRON¹は交換処理・情報処理・通信処理・ワークステーション等広い分野で共通に使用できる OS である。ソフトウェア流通性を促進するため、CTRON は基本 OS と拡張 OS の二階層による構成となっている。CTRON カーネルの機能は表 1 に示すようなサブセット単位に分割される。

筆者らは、NEC のオリジナル 32 ビットマイクロプロセッサ V70/V80 上に、C+M サブセット機能を持つ CTRON カーネルを実現した。本稿では V70/V80 CTRON の概要を紹介し、アーキテクチャ独立性を高め、UNIX² 上でも動作可能とした実現方式について述べる。

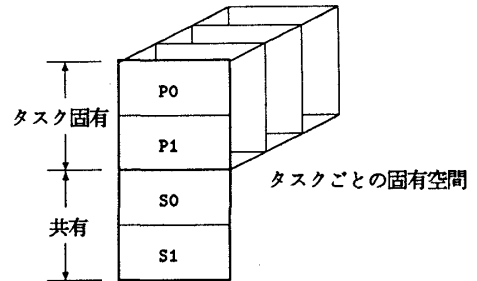


図 1: V70/V80 CTRON のメモリ空間

表 1: CTRON カーネルサブセット単位

サブセット単位名	機能内容	略記号	システムコール数
共通部	マイクロ	μC	44
	マクロ	C	84
タスク管理系及び タスク間同期通信系 選択部	タスク管理 タスク間同期通信 イベントフラグ メッセージボックス 割り込み管理 時計管理 メモリ管理	M	19
メモリ管理系選択部	タスク間同期通信 セマフォ シリアリリユーザブル 例外管理 統計情報管理	I	22

2 メモリ空間構成

V70/V80 の仮想空間は 4G バイトあり、これを 1G バイトの部分空間に分割して、アドレスの小さい順に P0, P1, S0, S1 空間と呼ぶ。

P0 空間 タスク固有空間

P0 空間は先頭からタスクのユーザーレベルスタック、タスク固有データとして使用する。

P1 空間 タスク固有空間:(将来の拡張のために予約)

S0 空間 共有空間

カーネルテキスト・データとコンフィギュレーションデータよりなる。コンフィギュレーションデータにより、カーネルを再構成をせずにカーネル資源のサイズ変更を可能としている。

S1 空間 共有空間

マップメモリ・共有ライブラリ・タスクテキストと

"An Implementation of CTRON Kernel on V70/V80 Microprocessors," *Akihiro Uchida,*Yukikazu Nakamoto,*Naoki Satou,*Masayuki Otaka,*Tadashi Koizumi,*C&C Common Software Development Laboratory,*1st Switching Network Systems Division,NEC Corporation

初期化データ・メモリプール領域・拡張 OS 用作業領域が置かれる。

共有ライブラリは、割り込みの中で走行する割り込みハンドラ・周期起動プロシジャと、全タスクから共通に使用される関数・データを置くために使用する。

マップメモリは IO 空間やデュアルポートメモリ・不揮発性メモリなど特殊用途の対象を仮想空間へマッピングするものである。

カーネルを UNIX 上で動作させるため、またデバッグのためのシンプルなモデルを提供するために、多重仮想空間を使用しないメモリモデルも実現している。

3 CPU 独立なインタフェース

CTRON カーネルを、CPU アーキテクチャに独立な OS コアとアーキテクチャ依存部分に分け、CPU アーキテクチャ依存コードを分離している。

OS コアとアーキテクチャ依存部のインタフェースを UNIX 上でシミュレート可能なものとする事で、単に他 CPU への移植性を高めるだけでなく、UNIX 上で OS を動作させることも可能とした。

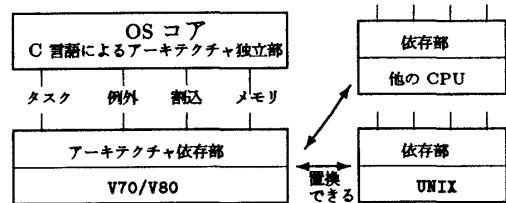


図 2: CPU 独立なインタフェース

OS コアのインタフェースは、機能的にはタスク・例外・割り込み・メモリ管理に分類される。これらの機能は、関数またはマクロ・データ構造の形で使用するか、あるいはアーキテクチャ依存部から OS コアのエン트리を呼び出させるために使用される。

¹CTRON は "Communication and Central TRON" の略称、TRON は "The Realtime Operating system Nucleus" の頭字略です。

²UNIX は米国 AT&T 社が開発しライセンスしています。

はじめに、一般的な OS で必要とされるインタフェースを挙げる。

- タスクスイッチ (関数)
- インタバルタイマコール (エントリ呼び出し)
- 割り込みハンドラ (エントリ呼び出し)
- 例外ハンドラ (エントリ呼び出し)
- ユーザレベルハンドラコール (関数)
特権レベルから関数をユーザレベルで呼び出す。
- ユーザレベルハンドラコールからの戻り (関数)
ハンドラを呼んだ時点の状態に戻る。
- 非同期トラップ設定 (関数)
- 非同期トラップ検出 (エントリ呼び出し)
システムコール・例外処理・割り込み、終了時に非同期トラップを検出し、ハンドラを呼び出す
- 仮想空間の確保 / 解放 (関数)
- 物理メモリのマッピングと解放 (関数)

3.1 CTRON 固有機能のためのインタフェース

CTRON カーネルでは待ち状態を中断して起動された例外ハンドラが終了したとき、中断した待ち状態に戻ることができる。これを実現するために次のインタフェースが必要となる。

- タスク実行状態のコピー (関数)
- コピーされたタスク状態へのスイッチ (関数)
コピーされた実行状態を使ってタスクスイッチすることで、元の待ち状態に復帰する。

周期起動プロシジャはクロックハンドラ内で呼び出され、実行が長引くとその実行中に呼ばれた別のクロックハンドラでタイムアウトが判定される。タイムアウトが判定された周期起動プロシジャを強制的に終了させるために、つぎのインタフェースが必要である。

- クロックハンドラ呼びだし場所の記憶 (関数)
- クロックハンドラ呼びだし場所のコピー (関数)
周期起動プロシジャ種別ごとの状態保存エリアにコピーする。
- クロックハンドラ呼びだし場所への復帰 (関数)
周期起動プロシジャを呼び出したクロックハンドラに復帰することで、周期起動プロシジャを中断する。

4 V70/V80 での実現と UNIX シミュレーション

ここに挙げた、OS コアを動作させるためのインタフェースは、メモリ管理を除いてどれも UNIX プロセス上でのシミュレーションが可能である。

タスク

V70/V80 はタスクコンテキストを保存 / 復帰する命令、`ldtask`、`sttask` を持っている。

UNIX では、アセンブラ命令によりフレームポインタ・レジスタをセーブ / ロードすることによりコンテキストスイッチを行なう。

インタバルタイマ

V70/V80 では、4ms 周期のインタバルタイマからの割り込みを使用する。

UNIX シミュレーションではアラームシグナルを使用する。

非同期トラップ

V70/V80 では特権レジスタの非同期トラップビットを立てておけば、`retis` 命令による割り込み / 例外からの復帰時にトラップハンドラが呼ばれる。

非同期トラップを持たない CPU、また UNIX でこの機能を実現するためには、割り込み / 例外からの戻り時に次の処理を行なう。

1. 戻り番地をトラップ検出ルーチンに書き換える。
2. タスクに戻る。トラップがかけられていればハンドラを呼ぶ。
3. 1 で書き換える前の場所に戻る。

タスク実行状態コピー

タスクスイッチ命令で保存された内容を待ち状態を管理する構造にコピーする。さらにスタックポインタをずらしてスタック内容を保存する。

UNIX では待ち状態管理構造に `setjmp`、`longjmp` で実行場所を保存する。

クロックハンドラ呼びだし場所への復帰

V70/V80 では割り込み発生時のスタックポインタを記憶し、タイムアウトが検出されたときは、スタックポインタを巻き戻すことで周期起動プロシジャを中断する。

UNIX では `setjmp`、`longjmp` を用いる。

5 アーキテクチャ依存部の規模

表 2 に CTRON カーネルプログラムのアーキテクチャ依存コード比率を示す。

表 2: アーキテクチャ依存コードの比率

種別	比率 (行数)
アーキテクチャ独立 C 言語	63%
アーキテクチャ独立ヘッダ	19%
V70/V80 依存 C 言語	8%
V70/V80 依存アセンブラ	7%
V70/V80 依存ヘッダ	3%
合計	100%
UNIX シミュレーション用	5%

多重仮想空間を使用しないモデル実現のためのコードが、アーキテクチャ独立のプログラム中に 12 箇所 `'#ifdef'` によって埋め込まれている。

6 おわりに

UNIX 上でシミュレート可能な CPU 機能インタフェースを定義することで、アーキテクチャ依存部を明確に切り分けることができた。V70/V80 CTRON カーネルは今後 CTRON 検定を受ける予定である。

参考文献

- [1] トロン協会 - CTRON 専門委員会編, " 原典 CTRON 大系 2 カーネルインタフェース", 1988.
- [2] 内田, "UNIX プロセスによる OS シミュレーション", 第 13 回 UNIX シンポジウム予稿集, 1989.
- [3] 佐藤他, "V70/V80 CTRON 例外ハンドラの実現", 情報処理学会第 39 回全国大会, 1989.