リアルタイムUNIXのモジュラーOS化検討

5C－1

落合真一、菅井尚人

三菱電機（株）情報技術総合研究所

1.はじめに

リアルタイム処理を扱うOSは、適用箇所の要求が多様なため、個々に異なる機能や実現上の制約を満たすことが必要となる。そのため、システムの各適用箇所に異なるOSを採用することになり、開発負荷が大きい。例えば図1のシステムの例では、次のような要求がある。

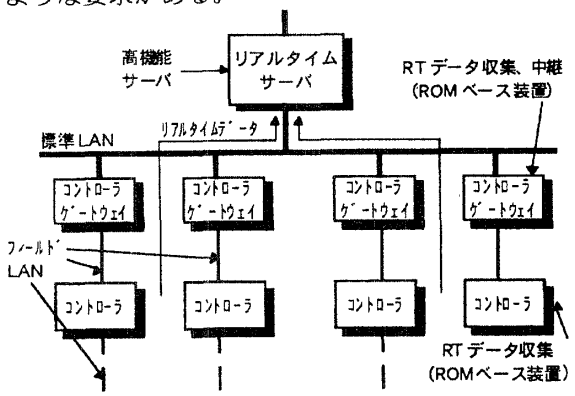

図1：リアルタイムシステムモデル

コントローラ：ROMベース実行、低コスト要求

ゲートウェイ：フィールドLAN/標準LAN両提供

リアルタイムサーバ：汎用機能、マンマシン制御

本稿では、産業制御システムへの適用を目指し開発中のリアルタイムUNIX "*moose*"をモジュール化することにより、コントローラなどの下位機種に適用範囲を広げ、統一OSによるシステム構築を可能とすることを目的とする。

2.開発課題と方針

リアルタイムUNIXのような大規模なOSをコントローラに適用するには次の課題がある。

(1) 適用箇所に合わせた機能選択性の提供

(2) ROM化可能なOS最小構成の実現

(3) OS使用資源の最適制御

(4) 起動時間の短縮

(1)、(2)の課題を解決するスケーラブルOSを実現する一つの方法として、マイクロカーネルをベースにしてOS機能を実現する方法がある。しかし、ここでは現状の"*moose*"のリアルタイム性、信頼性を維持し、短期開発を実現する面から、OSのモジュール分割により、課題を解決する方策を選択した。また、(3)のためにカーネルパラメータ変更機構の拡張、(4)のために起動機構の拡張を行う。

3.実現方法

3.1.OSのモジュール分割

各適用箇所の要求に対応可能にするために、プロセス管理、メモリ管理、I/O管理などの基本機能を提供するカーネルコアと、FS (ファイルシステム)モジュール、拡張機能モジュールに、OS機能を分割する(図 2)。カーネルに組み込むモジュールは、構成定義モジュールによって決定する。モジュール化により、リアルタイム性能を損なわないようにするため、各モジュールは使用する資源の排他制御を行い、マルチスレッド実行を可能に設計した。FSモジュールは、ファイル操作関数、バッファキャッシュ、FS依存データ構造体など、ファイルシステムに関係した全てを持つ。従来のUNIX互換FSの他に、新規にミニマムFS、優先度バッファキャッシュFS [1]を提供することにより、各構成要素の要求に応じたファイルシステムを選択可能にする。

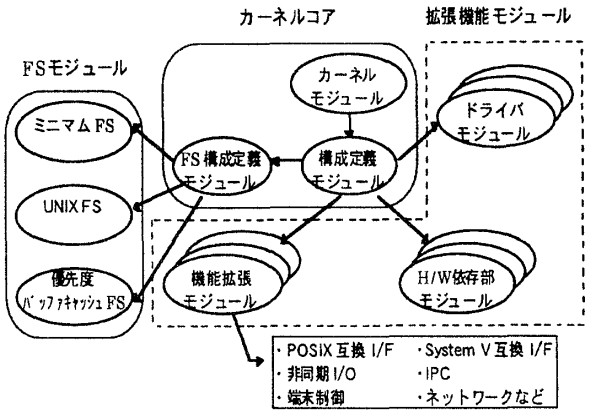

図2：モジュール構成

3.2.ROM構成用ファイルシステム

UNIXでは、ファイルシステムがカーネルの大きな部分を占めると共に、API提供の基となる。しかし、ROMベースの環境では従来のようなファイルシス

An Experimental Development of Modularized Real-Time UNIX
Shinichi OCHIAI, Naoto SUGAI
Mitsubishi Electric Corporation

テムは不要である。そこで、最小構成を実現するために、ファイルシステムとしての最小限機能を提供するミニマムFSを新規に作成することにした。
ミニマムFSは、ROM、RAM、システムバスなどのアドレス空間の一部を直接ファイルシステムとして扱う。ミニマムFS上のファイル構造は、事前連続領域割当てを基本にした単純な構造とし、バッファキャッシュレスで、fsckの実行を不要にする。
また、他のFSモジュールと同様に、カーネル内のFSモジュールI/Fに基づいて設計し、他のFSモジュールとの共存や置換えが可能とする。これらにより、最小構成においても、APIの変更はなく、従来とのバイナリ互換を保つ。ミニマムFSの使用により、図3のような構成のOSが構築可能である。

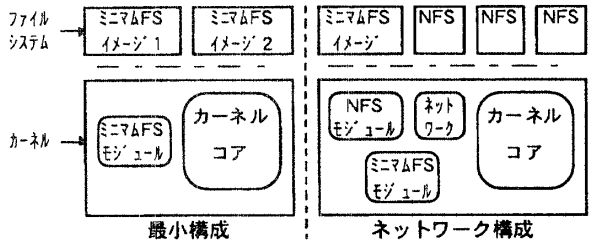

図3：ミニマムファイルシステムによるOS構成例

3.3.カーネルパラメータの起動時変更

カーネルが確保する資源の量は、カーネルパラメータによって決まり、これがOS使用メモリ量に大きな影響を与える。コントローラで行う処理は固定的であるので、パラメータ値を最適化することは有効である。しかし、従来のUNIXではパラメータ値の変更にはOS再生成が必要なため、システムの構成要素ごとにパラメータ値を変更することは、保守の面から困難であった。
この問題を解決するために、OS起動時にカーネルパラメータ値を動的に設定する機構を提供する。ブートデバイス上にパラメータ値を記述したファイルを用意し、次の動作を行う。

(1) ブートローダがカーネルローディング時に、指定されたパラメータファイルも読み込む。
(2) 設定された上限、下限値との比較より、パラメータ値の正当性を確認する。
(3) カーネル初期化ルーチンにより、指定されたサイズのデータ領域を拡張し、初期化を行う。
(4) ファイルが指定されなかった場合や、値が異常な場合はデフォルト値で領域を確保する。

これにより、パラメータファイルのみの変更によって、カーネルの使用メモリ量を最適化できる。

3.4.高速再起動

コントローラでは障害発生時に再起動により復旧を図るため、再起動時間に対する要求が厳しい。この要求に応えるために、高速再起動機構を提供する。
高速再起動時には、カーネルのロードやミニマムFSイメージの初期化は行わず、メモリ上のイメージを再利用して起動する。

(1) 初期起動時にカーネル内の初期化データのバックアップを作成する。
(2) S/W障害やリセット要求を受けた場合、バックアップデータを使って、カーネルデータの再初期化を行う。
(3) カーネルのエントリアドレスにジャンプすることによりOSを再起動する。

これにより、診断時間やカーネルロードの時間を削減した高速な再起動が実現できる。

4.評価

(1) OS構成
OSの使用メモリ量は、モジュール構成の選択により、従来の1/4にまで削減可能となった。

OS実行時の使用メモリ量

カーネル構成	使用メモリ量
最小構成	従来の約1/4
ネットワーク構成	従来の約1/2
従来互換(フルセット)構成	従来と同等

(コード+データ+実行時ヒープ+必須デーモンを含む)

(2) リアルタイム性能
モジュールI/Fのオーバヘッドによる性能低下は計測されなかった。これは、マイクロカーネル方式に対する優位点と判断する。
(3) 再起動時間
再起動時間は初期化すべきIOドライバの数により変わる。TTY、SCSI、LANなどの標準のIOドライバ構成では、高速再起動の場合、通常再起動の1/8の時間で起動可能となった。

5.おわりに

本モジュール化の検討により、現状のリアルタイムUNIXのスケーラビリティを下位機種方向へ広げる見通しが得られた。今後は実システムの要求に対応させた有効性検証を行う。

参考文献

[1] 菅井、落合：「リアルタイムUNIXのバッファキャッシュ管理方式」、情報処理学会第51回全国大会、1997