

## UNIX環境を搭載したリアルタイムOSの一構築法

## 2V-5

坪井俊洋、登丸浩、菅沢登、津村和政

沖電気工業株式会社

1. はじめに

リアルタイム (RT) UNIX の実現方向として、UNIX 環境も搭載可能としたリアルタイム OS (RT-OS) の一構築法について考察した。実験的試行として、MC-68010 上で動作する基本核 OS を開発し上位に UNIX 環境を拡張可能とする構造化・階層方式を採った。

2. 開発のねらいと設計方針

充実したプログラム開発環境と高速応答性信頼性という2つの要求を共存させることを目的とし、以下の方針で開発を行った。

## (1) リアルタイム応答の高速化

高い実時間性とプログラム動作の決定性を提供する。

## (2) ROM化の可能性

基本核 OS 部は、専用埋込型装置にも適用できる ROM 化可能な構造とする。

## (3) コンパクト性とモジュール階層化

基本核 OS 部のメモリ容量が小さいのみならず、個別製品に応じてシステムの最適構築が可能な階層化モジュール構造とする。

## (4) 高水準言語記述

基本的には、可能な限り C 言語にて記述し、他の計算機上への移植を容易にする。

## (5) 構造化 OS (マルチ OS の提供)

RT 環境において要求される処理性能を提供する基本核 OS の上に、ユーザ端末に必要な作業環境を満足させる OS (今回は UNIX) を構築可能な構造化

された OS とする。また、異なる OS がそれぞれ得意の分野を機能分担し、機能的に相互間の独立性を保つ。

## (6) UNIX との親和性

次に示す各レベルにおいて UNIX との親和性を実現する。

## \* 開発環境

## \* ファイルシステムの互換性

## \* 拡張 OS としての UNIX 搭載

## (7) マルチプロセッサ対応

マルチプロセッサ化に対しても、拡張可能なプロセッサ間通信機能をシステムプリミティブとして提供する。

3. 構成と機能モデル

図1に、ソフトウェアの階層構成を示す。この3つに階層化された構造により第2・第3階層のプロセス群の選択及び追加が容易にでき、ユーザの要求する機能に応じたシステムを自由に構築することが可能となる。以下に、代表的なモデルを示す。

## (1) Small configuration model

- 単一論理空間システム
- コンパクト 高速リアルタイム応答
- ソフトウェア構成 (基本機能)

## \* カーネル

## \* I/O ドライバ群

## \* ファイルマネージャ

## (2) Medium configuration model

- 多重論理空間システム
- 高機能リアルタイム OS
- ソフトウェア構成 (拡張機能)

## \* プロセスマネージャ

- \* メモリマネージャ (含スワップ)
- \* スケジューラ

(3) Large configuration model

- UNIX環境の提供
- \* Cライブラリ
- \* シェル
- \* UNIXエミュレータ

4. 実現方式

(1) 基本核OSの機能

基本核OSは、多種の拡張OSを実現するために必要最小限のシステムプリミティブに限定した。基本機構としてメッセージパッシングを採用し、プロセスへの非同期割込み通知機能を設ける事で拡張OS側でのシステムプリミティブの拡張を可能にした。

(2) スケジューリング方式

プロセスの振舞に決定性を与える固定優先度・事象駆動型のディスパッチメカニズム (RT環境) と、系内リソース利用の調和を図れる動的優先度によるスケジューリング (TSS環境) の二方式を装備した。

(3) I/Oドライバ

プロセスへの非同期割込み通知機能を

外部I/O割込みに拡張することで、I/Oドライバをプロセスとして開発可能とした。

(4) ファイルシステムの実現

UNIXとRTOS相互間でファイルメディアの直接交換を可能とした。更に、大容量ファイルの高速アクセスの為にcontiguousファイルを導入した。

(5) UNIX環境の搭載手法

RTOSの拡張として、UNIXプログラミングインタフェース及び、会話インタフェースを実現し、RT環境で動作するソフトウェアの開発を可能にする。このUNIXエミュレータはユーザプロセスのコンテキストで動作し、(1)の基本機能の組み合わせによりUNIXシステムコールを提供する。

5. むすび

UNIX環境とRT環境を一体化させるための一インプリメンテーションについて述べた。本方式は、上位OSを実現するための必要最小限の機能を抽出した基本核OSを実現し、システム全体としての統一性を保ちつつ複数のOS環境を実現可能としている。

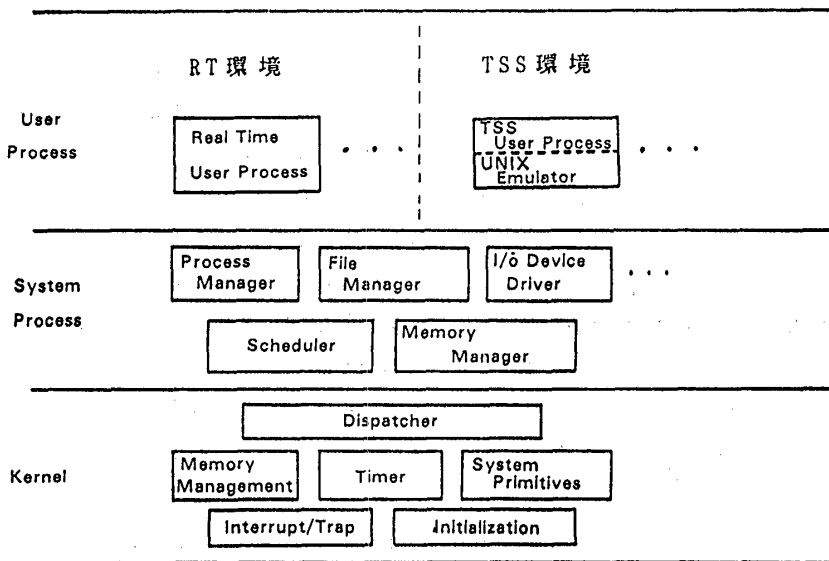


図 1 Software Hierarchy