

木村信二\*、新井利明\*、佐藤雅英\*、大島訓\*、梅都利和\*\*

\* (株)日立製作所システム開発研究所、\*\* (株)日立製作所情報機器事業部

### 1. はじめに

1台のマシン上で複数のオペレーティングシステム(OS)を共存させるナノカーネル方式の開発を進めている。Windows NTともう一つのOSを共存させ、複数OS間の機能連携、機能補完を可能とすることによって、システム全体の信頼性、拡張性を確保できることを既に提案した。[1][2]

一方、Linuxは広く普及しているオペレーティングシステムであり、柔軟な構成を取ることができるためPCサーバから組み込みシステムまで利用されている。しかし、従来の標準OSと同様に拡張性やリアルタイム性が不十分な点がまだ残っている。本稿では、Linuxオペレーティングシステムの機能を拡張する手段として、Linux対応ナノカーネルについて報告する。特に、従来のWindows NT対応ナノカーネルとの実装上の相違点について述べる。

### 2. Linux対応ナノカーネル

ナノカーネルは、ハードウェアとOSの間に位置するソフトウェアであり、ホストOS(第1OS)からハードウェア制御を横取りし、ゲストOS(第2OS)の実行環境を作り出す(図1)。

ナノカーネルが提供する機能を以下に示す。

#### (1) 資源管理機能

計算機資源を分割し、各OSおよびナノカーネル自体に割付ける機能である。メモリおよびI/O機器は指定されたOSで占有し、CPU、タイマはOS間で共用する。

#### (2) OS間通信機能

異なるOS上のプロセス間での直接通信を可能とす

る機能であり、共用メモリ、メッセージパッシング、プロセス同期機能を提供する。これにより、OS間の機能連携、機能補完が可能となる。

#### (3) 障害監視、回復機能

ナノカーネルの特徴機能の一つであり、OSに障害が発生した際に、その状況を他OSから監視し、障害が起きたOSのみの再起動することができる。これにより、OS障害の影響を最小限に限定できる。

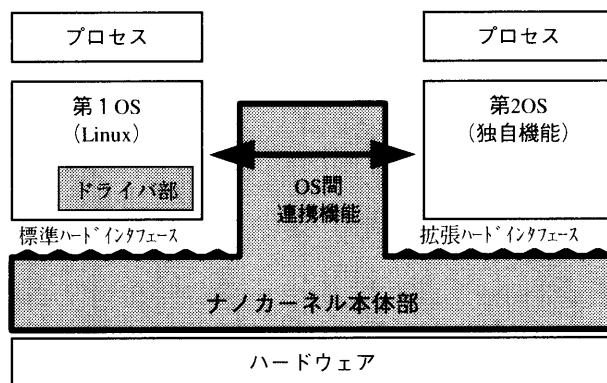


図1 Linux 対応ナノカーネルの基本構造

### 3. 実装方式

以下では、LinuxをホストOSとして動作するナノカーネルの実装について述べる。

#### 3.1 カーネルモジュール化

ナノカーネルは各OSの共通処理として動作するため、各OSの仮想空間上の同じアドレスに配置する必要がある。しかし、ゲストOSにLinuxを動作させる場合、仮想アドレス空間上のOS空間がホストと重なり、ホストLinuxに組み込まれたナノカーネルのアドレスを、ゲストOS側で確保することは困難である。そこでナノカーネルを各OS空間上の同じアドレスに配置するため、Linuxのカーネルモジュール機能を利用した。具体的には、以下の手順で実装した。

(1) ナノカーネルを、ホストOSとのインタフェース処理を実行するドライバ部と、ナノカーネル本体部の2つに分けて構成する。

“An Implementation of the multi-OS environment using Nano-Kernel Method”,

Shinji Kimura\*, Toshiaki Arai\*, Masahide Sato\*, Satoshi Oshima\*, Toshikazu Umedu\*\*

\* Systems Development Laboratory, Hitachi, Ltd.

\*\* Mechatronics Systems Division, Hitachi, Ltd.

(2) ナノカーネル本体部のオブジェクトをLinuxが提供するカーネルモジュール機能により、ホストとゲストOSに共通したアドレス空間に再配置する。

(3) 再配置したオブジェクトを、ホストOSに組み込んだナノカーネル・ドライバ部を經由してLinuxの仮想空間にローディングする。

また、カーネルモジュール化は、カーネル自身を変更する実装に比べ、カーネルの内部情報へのアクセスなどの機能が制約されるが、カーネルの再コンパイルを必要とせずに導入できるという利点がある。

### 3.2 ホストOSの再起動方式

ナノカーネルは物理メモリを、ホストOS領域、ゲストOS領域及び管理領域に分割し、それぞれのOSに割り当てる。各OSを物理メモリ上で隔離することによって、OSの障害が他のOSに伝播することを最低限に抑える。ナノカーネル本体部は、前記ローディング時に、物理メモリ上で各OSから独立した管理領域内に配置することによって、各OSに依存せずに行うことが可能とした。これにより、ホストOSに障害が発生してもナノカーネル自身の処理は継続でき、さらにナノカーネルからホストOSのみを再ローディングする機能も実装した。

### 3.3 障害検知

ナノカーネルは、前述のように、1つのOSに障害が発生しても、他のOSの処理を継続できる。OSの障害を検知するため、2つの障害検知処理を実装した。1つはソフトウェア割り込みの命令を使った方法であり、OSのエラー処理部に割り込み命令を埋め込み、この割り込みをナノカーネルで検知する。2つ目は、ソフトウェア・ウォッチドックタイマー機能である。ナノカーネルが提供する共有メモリを使い、2つのOS上のプロセスが定期的な通信によって、OSの異常を監視する。

### 3.4 メッセージ通信

2つのOS上のアプリケーションはナノカーネルのOS間通信機能を使用し、互いに通信を行うことができる。しかし、ナノカーネルの専用インタフェースを使用する必要があるため、既存のアプリケーションでは通信処理を改造する必要があった。このた

めナノカーネルのOS間通信機能の上位レイヤとしてTCP/IPプロトコルへのルータ機能を実装した(図2)。Linuxの標準機能を使用し、Ethernetドライバの通信データをルータ(アプリケーションプログラム)で受け取り、ナノカーネル専用インタフェースに変換する。このルータ機能を用いることにより、OS間でtelnetやftpコマンドなどの既存のプログラムを使った通信が可能になる。

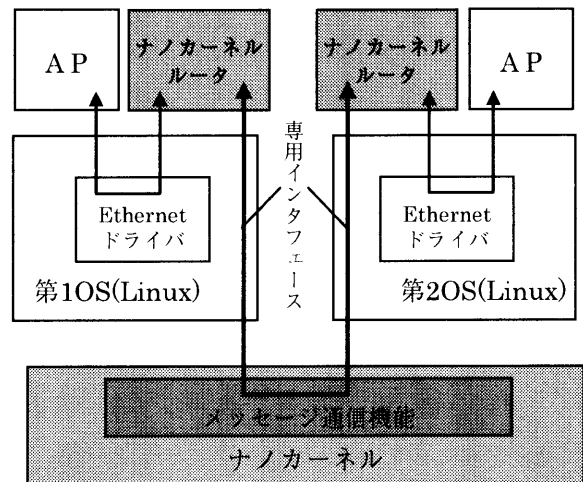


図2 TCP/IPプロトコルによる通信

## 5. おわりに

以上、ナノカーネル方式のLinuxオペレーティングシステムへの適用について述べた。標準OSの利点である豊富なソフトウェア環境に加えて、OSを改造することなく独自の機能を拡張することができる。また、複数のOSが協調してシステム全体の信頼性向上を図ることができる。

## 参考文献

- [1] 新井他、「ナノカーネル方式による異種OS共存技術「DARMA」の提案」, 情報処理学会, 第59回全国大会 講演論文集(1), pp.139 - 140, 1999.
- [2] 佐藤他、「ナノカーネル方式による異種OS共存技術「DARMA」の実装」, 情報処理学会, 第59回全国大会 講演論文集(1), pp.141 - 142, 1999.

Windows NT(R)は、米国Microsoft Corporationの米国および他の国の登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標あるいは商標です。