マイクロカーネル Lavender における IPC 機構とデバイスドライバの構成

2F−3

豊岡 明† 佐脇 秀登† 芝 公仁† 毛利 公一† 大久保 英嗣††
†立命館大学大学院理工学研究科 ††立命館大学理工学部情報学科

1 はじめに

分散環境への対応は，現在のコンピュータの利用形態として必須であると考えられる．Lavender[1] では，ネットワーク透過な IPC (Inter Process Communication) のインタフェースを提供することで，これに対応する．また，従来のマイクロカーネルでは，カーネルとシステムサーバ間における IPC のオーバヘッドの大きさが問題になっている．Lavender では，同一ノード内での IPC を，共有メモリにより実行時に効率化する．

デバイスドライバにおいても，Lavender はネットワーク透過なインタフェースを提供する．これにより，ネットワークによるデバイスの共有が容易に実現できる．また，デバイスドライバのインタフェースは，デバイスの持つ機能に密着した手続きから構成されており，オーバヘッドを小さく抑えられる．デバイスドライバの安全性の点では，デバイスドライバをユーザモードで動作させることにより，不正なハードウェアへのアクセスをカーネルが検出できる．また，PC カードや USB に代表される，活線挿抜可能なデバイスへの対応が求められている．Lavender のデバイスドライバは，システムサーバプロセスおよび LKM (Loadable Kernel Module) として実現され，動的な追加および削除を可能としている．

以下，本稿では，2 章で IPC 機構，3 章でデバイスドライバについて述べ，4 章で本稿のまとめを述べる．

2 IPC 機構

Lavender の 提供する IPC には，共有メモリによる IPC と，ポートを基本とした IPC の 2 種類がある．どちらの IPC においても，Lavender はネットワーク透過なインタフェースを提供する．これにより，同一ノード内での IPC と他ノードにまたがる IPC を，ユーザが区別する必要はない．

2.1 IPC 機構の構成要素

1. ポート …IPC において通信相手を特定する識別子である．通信相手はポート番号のみで区別されるため，通信相手がカーネル，システムサーバ，ユーザプロセスのどれであるかは，ユーザから隠蔽される．同様に，通信相手がどのノードに存在するのかも隠蔽される．

2. メッセージ … 1 回の IPC で転送される，任意の大きさを持つデータのブロックである．

3. **IPC サーバ**…ポートの割り当てと解放，メッセージの送受信，メッセージやポートの状態取得などの，基本的なインタフェースを提供する．

4. **IPC ライブラリ** …IPC サーバへのインタフェースや，ネームサーバへのインタフェースを提供する．また，IPC の効率化をユーザから隠蔽する．

5. ネームサーバ … ポート番号とポート名の対応は，ネームサーバによって管理される．ネームサーバはシステムサーバとして動作し，ネーミングポリシを決定する．ユーザプロセスは，通信相手のポート番号を得るために，ポート名を指定してネームサーバに問い合わせる．

2.2 IPC の処理方式

Lavender では，カーネルとユーザプロセス間，およびユーザプロセス間でメモリを共有できる．送信プロセスと受信プロセスの同期を取り，共有メモリにメッセージを読み書きすることで，コピーの発生しない通信を行える．

さらに，Lavender では，ポートによる IPC を使用することで，任意の大きさのメッセージを送受信できる．同一ノード内でのポートによる IPC では，IPC サーバと IPC ライブラリにより，実行時に通信の効率化が行われる．この効率化には共有メモリを使用する．以下に効率化の手順を示す．

1. IPC の開始時に，通信相手が同一ノード内に存在するかを，IPC ライブラリが IPC サーバに問い合わせる．

2. 同一ノード内に存在する場合は，通信相手の IPC ライブラリとの間に共有メモリを確保する．また，効率化を行うことを IPC サーバに通知する．

3. メッセージの送信時には，共有メモリ内にメッセージを作成し，送信の要求を IPC サーバに出す．

IPC Mechanism and Device Driver in Lavender Micro Kernel
Akira Toyooka†, Hideto Sawaki†, Masahito Shiba†, Koichi Mouri† and Eiji Okubo††
†Graduate School of Science and Engineering, Ritsumeikan University
††Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University

4. 受信側 IPC ライブラリでは, 共有メモリからメッセージを取り出し, ユーザプログラムに渡す.

5. IPC の終了時には, 共有メモリを解放し, IPC サーバに通信の終了を通知する.

3 デバイスドライバ

Lavender のデバイスドライバは, 以下の特徴を持つ.

1. **ポリシを持たないデバイスドライバのインタフェース** … Lavender では, デバイスの抽象化はシステムサーバが実現する. よって, システムサーバのポリシによって, デバイスがどのように抽象化されるかは異なる. そのため, デバイスドライバのインタフェースでは, 低レベルな抽象化しか行わない.

2. **クラスによるデバイスの分類** … デバイスを, その特性に応じたクラスによって分類する. 各クラスは, デバイスの持つ機能に対応した, 関数の集合を定義したものである. システムサーバは, 各クラス毎に定義されたインタフェースを用いて入出力を行う.

3. **分散環境への対応** … デバイスドライバは, 他のシステムサーバやカーネルとプロセス間手続き呼び出しを用いて通信する. この構成により, 他ノードのデバイスドライバと自ノードのデバイスドライバを同じインタフェースで使用できる.

4. **オーバヘッドの軽減** … デバイスドライバの呼び出しにはプロセス間手続き呼び出しを用いる. プロセス間手続き呼び出しは, 2 章で述べた IPC を使い, 他のプロセスの手続きを呼び出す機能を実現したものである. プロセッサが実行権限の移行を伴う手続き呼び出し命令をサポートする場合, このプロセス間手続き呼び出しは効率化される.

5. **ハードウェア資源の管理** … デバイスドライバはハードウェア資源管理部に, 使用する I/O ポート, 物理メモリアドレス, 割り込みなどの割り当てを要求し, 許可されてから使用する.

6. **動的なハードウェア構成の変更への対応** … システムの動作時における, デバイスドライバの動的な追加および削除が可能である.

3.1 デバイスドライバの構成要素

Lavender のデバイスドライバは, 以下の要素から構成される.

1. **ユーザモードドライバ** … ユーザモードで動作するシステムサーバプロセスとして実現される. 割り込み応答性能はカーネルモードドライバに比べ劣るが, より安全なデバイスドライバとして実現される.

2. **カーネルモードドライバ** … LKM として実現される. 安全性の面で従来のデバイスドライバと同様の問題を持つが, 高速な割込み応答速度を持つ.

3. **ハードウェア資源管理部** … I/O ポートや物理メモリなどのハードウェア資源を, 要求に応じてデバイスドライバに割り当てる.

4. **デバイスドライバライブラリ** … ドライバのクラス毎に用意される. デバイスドライバ用ライブラリと, 対応するユーザ用ライブラリがある.

3.2 ドライバの実現法

Lavender のデバイスドライバは, I/O 権限のあるユーザプロセスとして実現される. デバイスドライバはユーザモードで動作するため, ハードウェア資源管理部に許可されたハードウェア資源しか使用できない. また, 不正なハードウェア資源へのアクセスは, カーネルによって検出できる. これにより, デバイスドライバのバグや, 資源の競合によるシステムの障害を防止できる.

高速な割り込み応答速度を要求されるデバイスや, カーネル内で I/O 処理をする必要のあるデバイスの場合は, カーネルモードで動作するデバイスドライバが必要になる. これに対応するため, Lavender では, デバイスドライバを LKM としてカーネル内にロードする機構を持つ. ただし, カーネルモードドライバは, 安全性の面で従来のデバイスドライバと同様の問題を持つ.

4 おわりに

本稿では, Lavender における IPC 機構とデバイスドライバの構成について述べた.

IPC 機構は, ネットワーク透過なインタフェースを提供する. また, 共有メモリを利用した, プロセス間通信の実行時における効率化を実現する.

デバイスドライバの構成では, IPC を用いることでネットワーク透過なデバイスドライバのインタフェースを実現している. デバイスドライバのインタフェースの点では, デバイスの抽象化をシステムサーバで実現することで, マイクロカーネルとしての柔軟性と, オーバヘッドの軽減を実現している. デバイスドライバの安全性の点では, デバイスドライバをユーザモードで動作させることにより, 不正なハードウェアへのアクセスをカーネルが検出できる. また, システムサーバおよび LKM としてデバイスドライバを実現することで, システムの動作時における追加と削除を可能としている.

参考文献

[1] 芝 公仁, 佐脇 秀登, 豊岡 明, 毛利 公一, 大久保 英嗣: "マイクロカーネル *Lavender* の構成", 情報処理学会研究報告 97-OS-75, pp. 7-12, 情報処理学会 (1997).