

携帯端末向け小型 IPC の RT-Mach への実装と評価

5C-10

黒岩 実 石井 秀浩 高野 陽介 横田 実
NEC C&C 研究所

1 はじめに

近年、携帯端末や組込システムといったハードウェア制約の大きなシステムに対して、連続メディア処理やインターネットアクセスといった汎用機と同等の機能を持つアプリケーションが求められている。従って、携帯端末の OS には、小型であることと同時に、それらのアプリケーションをサポートする高度な機能が要求されている。我々は、リアルタイム Mach(以下、RT-Mach)[1]をベースとして、メモリや二次記憶を有効利用する機能を備えた、小型の携帯端末向き OS の構築を目指している [2]。

本稿では、カーネルにおけるメモリ消費量の低減を目的として、携帯端末に適した小型の IPC を提案し、RT-Mach 上での実装、評価について述べる。

2 携帯端末向き IPC の構成

携帯端末用の IPC は、メモリ制約から小型である必要があるとともに、リアルタイム機能や効率的なデータ転送といった高度な機能も要求される。

それに対して、既存の組込システム用 OS が提供する簡単なメッセージ送受信機能は、小型ではあるものの、リアルタイム機能をはじめ機能的には不十分である。一方、RT-Mach などの汎用 OS が提供する IPC は、様々な通信形態をサポートするために豊富で柔軟な機能を提供するものの、固定的なアプリケーション構成を持つ携帯端末では使用しない機能も含まれ、必要以上のメモリを消費してしまう。

そこで我々は、汎用 OS が提供する豊富な通信機能を、タスク間の通信に必要な最小限の基本機能と、個別に組み込み可能な複数の拡張機能とに分離し、必要な機能をカーネル構築時に選択することによって携帯端末ごとに最適化可能な IPC を提案する。

2.1 基本機能

タスク間通信に必要な最小限の基本機能として、グローバルな名前での識別される通信路と、通信路に対するメッセージ送受信機能を提供する。

Implementation and Evaluation of Shrunked IPC for Handy Computers

Minoru KUROIWA, Hidehiro ISHII, Yosuke TAKANO and Minoru YOKOTA

C&C Research Laboratories, NEC Corp.

4-1, Miyazaki, Miyamae-ku, Kawasaki, Kanagawa 216, Japan

2.2 拡張機能

携帯端末のアプリケーション構成に応じてカーネル構築時に選択可能な拡張機能として、

- 通信路の保護機能
- 通信路の状態変化の通知機能
- ゼロコピー通信機能
- リアルタイム通信機能
- タスクにローカルな名前空間

を提供する。以下、携帯端末で想定される様々なアプリケーション構成が、どの拡張機能を組み込むことで実現できるかを説明する。

(a) プロトタイピング

デバッグが完了した信頼性の高いシステムでは、通信路が不正に利用されることはなく、また、通信路の構成は固定的であるため、通信路の保護機能や通信路の状態変化の通知機能は必ずしも必要ではない。

それに対して開発途中のシステムは、プログラムのバグによる通信路の不正利用や、タスクの異常終了といった通信中の障害に対して、より強固に構築する必要があるため、ケーバリティ制御による通信路の保護機能と通信路の状態変化の通知機能を組み込む必要がある。

(b) 連続メディア処理

動画や音声といった連続メディアを再生する携帯端末では、送受信タスク間での優先度継承などのリアルタイム通信機能を組み込むことによって、なめらかな再生が可能となる。

また、連続メディア処理では大量のデータを扱うため、ページテーブル操作を用いたゼロコピー通信機能を組み込むことによってオーバーヘッドを低減できる。

(c) オブジェクト管理

オブジェクトを扱うサーバ(例えば、ファイルサーバ)では、不要になったオブジェクトのメモリ領域を解放するために、オブジェクトを参照カウンタなどによって管理しなければならない。その場合、通信路の状態変化(この場合、送信タスクの消滅)の通知機能を組み込み、それぞれのオブジェクトに通信路を割り当てることによって、オブジェクト管理を効率的かつ簡単に実装できる。

(d) 通信モニタリング

性能評価やデバッグのために送受信タスク間に別のタ

スクを挿入して通信を監視する場合、グローバルな名前空間ではアプリケーションを変更する必要が生じる。その場合、タスクにローカルな名前空間を拡張機能として組み込むことによって、名前と通信路の対応をタスクごとに独立させることができるため、アプリケーションを変更せずに実現できる。

3 RT-Mach への実装と評価

3.1 実装方法

RT-Mach の Mach IPC[3] と RT-IPC[4] は上記の基本機能と拡張機能を全て包含している。そこで、基本機能に相当するポートを介したメッセージ送受信以外の機能を、拡張機能の単位で選択的に組み込めるようにモジュール化することによって、提案する携帯端末用 IPC を実装する。

なお、実装上の制約として、RT-Mach では通信路の保護機能とタスクにローカルな名前空間の 2 つの拡張機能はポート権によって同時に実現されるため、単独での組み込みは行わない。

3.2 メモリ消費量の評価

携帯端末用 IPC と Mach IPC について、メモリ消費量の下限値である静的メモリ消費量と、アプリケーション構成に応じて変動する動的メモリ消費量について比較する。

まず、静的メモリ消費量の比較を図 1 に示す。Mach IPC の静的メモリ消費量が 60 キロバイトであるのに対し、基本機能だけからなる携帯端末用 IPC では 30 キロバイトに半減できた。

また、今回は変更を行わなかったが、組み込む拡張機能に応じて MIG(Mach Interface Generator) の出力を最適化することによって、さらに 8 キロバイト程度の小型化を実現できる。

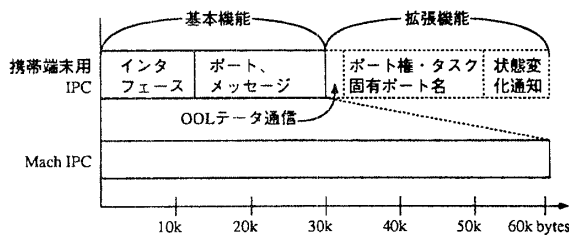


図 1: 携帯端末用 IPC の静的メモリ消費量

次に、ポートをひとつ生成するごとに消費される動的メモリ消費量の比較を表 1 に示す。基本機能だけからなる携帯端末用 IPC を Mach IPC と比較すると動的メモリ消費が 36% 低減できるが、これは 1000 個のポートを生成する場合には 48 キロバイトに相当する。RT-Mach では、スレッドやデバイスといったカーネル資源に対し

て個別のポートが割り当てられるため、ポートの使用量が増加する傾向があるが、そのような場合にもカーネルのメモリ消費量を低減できる。

表 1: 1 ポート生成時の動的メモリ消費量

Mach IPC	基本機能
132 バイト	84 バイト

4 課題

4.1 Mach IPC との互換性の維持

現在実装中の携帯端末用 IPC では、カーネルインタフェースのシンタックスを Mach IPC と共通化し、該当する機能が存在しない引数を無視することによって、ある程度の互換性を実現している。しかし、携帯端末用 IPC を使用するアプリケーションを Mach IPC の環境で開発する場合には、本来不必要なポート権に関する設定などを記述する必要がある。今後、ライブラリあるいは MIG レベルで十分な互換性を提供することによって、そのような手間を解消することが課題である。

5 まとめ

本稿では、メッセージ送受信に必要な最低限の基本機能だけから構成でき、カーネル構築時に必要に応じて拡張機能を組み込み可能な、携帯端末用の小型 IPC を提案し、RT-Mach 上で評価した。その結果、RT-Mach が提供する Mach IPC と比較して、基本機能だけの場合に静的メモリ消費量を 50%、動的メモリ消費量を 36% 低減できることがわかった。Mach IPC は RT-Mach で最もメモリ消費の大きな機能であるため [2]、その効果は大きいといえる。今後は、ライブラリや MIG による Mach IPC との互換性維持が課題である。

参考文献

- [1] H.Tokuda, T.Nakajima and P.Rao: "Real-Time Mach: Towards a Predictable Real-Time System," USENIX Mach Workshop, pp.73-82 (1990).
- [2] 黒岩、石井、高野、横田: "MKng プロジェクトにおける携帯端末サポート," 第 53 回情報処理学会全国大会論文集, 5B-5 (1996).
- [3] R.P.Draves: "A Revised IPC Interface," Proceedings of the USENIX Mach Conference, October (1990).
- [4] T.Kitayama, T.Nakajima and H.Tokuda: "RT-IPC: An IPC Extension for Real-Time Mach," In Proceedings of the Winter 1992 USENIX Conference, pp.91-104, Sep. (1993).