

ハイブリッド OS システムを用いた車載メディアプレーヤーの高速起動方式 A Fast Boot Method for In-Car Media Players Based on a Hybrid-OS System

飯野 晋[†] 江角 憲治[†] 野口 正雄[†] 楠 恵明[†] 東 辰輔[†]

Susumu Iino Kenji Esumi Masao Noguchi Yoshiaki Kusunoki Shinsuke Azuma

1. はじめに

国内の映像組込機器では、 μ ITRON に代表されるリアルタイム OS(RTOS)が多く使用されてきた[1]。RTOS は起動時間が速く、リアルタイム性が求められるハードウェア制御のソフトウェア実装が容易であるが、ネットワーク機能などの拡張ライブラリが乏しいという問題があった。一方、近年では映像組込機器に対する多機能化要求があり、2000 年頃からは機能拡張ライブラリが充実した汎用 OS である Linux が採用される例が増えている[2][3]。Linux ではカーネルやデバイスドライバに加え、必要に応じた機能拡張ライブラリの導入に伴ってプログラムサイズが大きくなるため、起動処理時間が RTOS に比べて遅くなる傾向にある。これまで、RTOS と汎用 OS それぞれの特徴(表 1)を補完する提案として、リアルタイム Linux や、RTOS と汎用 OS を組み合わせたハイブリッド OS システムが提案されている[4][5][6]。

Linux が採用される映像組込機器には、例えば車載メディアプレーヤーがある。車載メディアプレーヤーでは、DVD(Digital Versatile Disc)の再生機能を搭載することが多く、RTOS 上で動作する DVD 再生ソフトウェアで実現してきたが、近年はこれに加えて BD(Blu-ray Disc)再生機能、SD カードや USB メモリ等の記憶デバイス読み出し、インターネット接続等への対応が求められており、また多機能化に伴う UI の高度化も必要となるため、ファイルシステム、グラフィックス、ネットワークスタック等の機能拡張ライブラリが豊富な Linux を採用することが望ましい。

一方で、車載環境における映像組込機器では、エンジン停止状態からシステム起動(コールドスタート)において、起動画面の出力、コンテンツの再生開始等への遷移時間に対する要求が厳しく、Linux を採用するシステムでは、実現が困難な場合がある。

上記課題の解決には、先に紹介したハイブリッド OS システムの適用が一つの解となる。筆者らは、車載環境に適用可能なコールドスタートに配慮した、ハイブリッド OS システムを用いた車載メディアプレーヤーの高速起動方式を提案する。

表 1. RTOS と汎用 OS のメリット・デメリット

| | RTOS | 汎用 OS |
|-----------|------|-------|
| 起動時間 | 速い | 遅い |
| リアルタイム性 | ○ | △ |
| OS データサイズ | 小 | 大 |
| 機能拡張ライブラリ | 乏しい | 豊富 |

2. 提案方式の概要

本章では、提案方式で採用するハイブリッド OS システムの概要と、必要な内部処理の概要について説明する。

ハイブリッド OS システムを用いて車載メディアプレーヤーを構築するメリットは、(a)汎用 OS に含まれる機能拡張ライブラリの利用と、RTOS 上に構築したソフトウェア資産としてのミドルウェア(MW)を流用が両立できること、(b)RTOS の起動の速さを利用して、起動からコンテンツの再生開始までの時間を高速化できることである。

2.1 システムの基本構成

前節で述べた 2 つのメリットのうち、まず(a)を生かすハイブリッド OS システム構成を検討するために、機能を次の 3 つに分解する。

- (ア) RTOS 上に開発済みの MW 実行及び、デコーダ制御等のリアルタイム性が必要な処理を実行するシステム
- (イ) ネットワーク処理や Java 仮想マシンの実行、GUI 処理等、汎用 OS の機能拡張ライブラリを利用した処理を実行するシステム
- (ウ) 上記(ア)(イ)のシステムを制御するためのアプリケーションを実行するシステム

言うまでもなく、上記(ア)の役割には RTOS システム、(イ)の役割には汎用 OS システムを割り当てるのが相応しいが、(ウ)の役割をどちらに割り当てるべきかが問題になる。これを次節で検討する。

2.2 高速起動のための構成

2.1 節の構成において、上記(ウ)の役割を RTOS システムに割り当てる方式は文献[6]に示されており、(b)の通り起動を高速にできる。一方、機能拡張ライブラリ、特に汎用グラフィックスライブラリを GUI 開発に使用できるという点において、上記(ウ)の役割を汎用 OS システムに割り当てることのメリットも大きいと言える。

したがって、両 OS のメリットを享受するためには、(ウ)の役割を担うための機能を両 OS のシステムに実装するとともに、次のように(ウ)の機能の実行を途中で交替する方法が適当である。

- ・起動開始～汎用 OS 起動まで：

RTOS システムが一時的に(ウ)の役割を担当する (この間メインシステムは起動処理中)

- ・汎用 OS 起動後：

汎用 OS システムが(ウ)の役割を担当し、RTOS システムは汎用 OS システムから制御される

すなわち、大枠においては汎用 OS に(ウ)の役割を割り当てる一方、RTOS システムにも起動画面の出力、ユーザーからの再生開始要求等の操作が受付可能なような、機能制限された GUI 及び制御システムを持つ簡易アプリの機能を持たせ、汎用 OS の起動前のみ(ウ)の役割を担当させることを考える。

提案方式では上記を基本方針とし、汎用 OS システム(メインシステム)の起動後、RTOS システム(サブシステム)からメインシステムへシステム状態に関する情報

[†] 三菱電機株式会社 先端技術総合研究所

(システム状態情報)を引き継がせるとともに、操作要求信号の送信先をサブシステムからメインシステムへ切り替え、さらにサブシステム側 MW を制御する主体を簡易アプリからメインシステム側のアプリ (フルアプリ) へ切り替える処理を設ける。本論文ではこの処理を特に権限委譲と呼ぶこととする。

権限委譲を行うためには、ハードウェアは、操作要求信号の送り先を 2 つの CPU (CPU-S, CPU-M) のそれぞれに排他的に切り替える仕組みと、各 CPU 間で通知送信を行うための CPU 間通信の仕組みと、システム状態情報を引き継ぐための共有メモリ領域が必要となる。これらを実現するハードウェア構成を図 1 に示す。

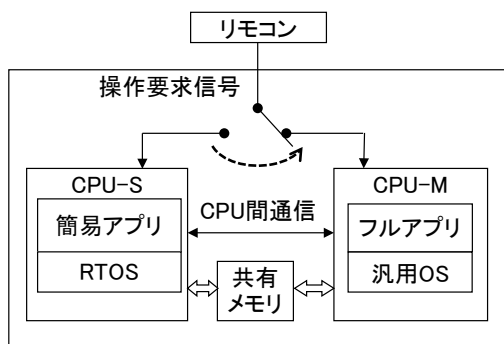


図 1. ハードウェア構成

また、ソフトウェアは、RTOS 側 MW の制御を簡易アプリと汎用 OS 側の MW のどちらにするかを切り替える処理と、制御主体の切り替え時にシステム状態情報を共有メモリで引き継ぐ処理が必要となる。これらを実現するために提案するソフトウェア構成概要を図 2 に示す。

引き継ぐべきシステム状態情報の内容は、対象システムに依存する。具体例に関しては 3 章にて述べる。

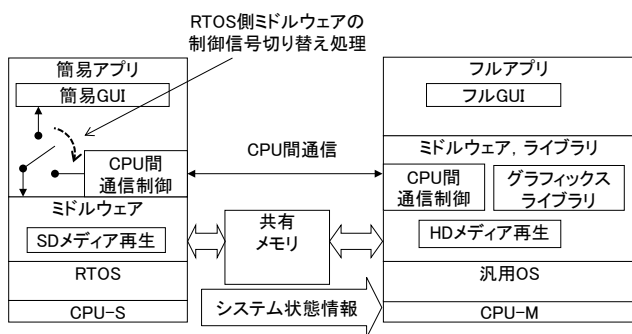


図 2. ソフトウェア構成概要

2.3 高速起動処理の概略

前節までで述べた構成を利用した高速起動方式の処理は、以下の流れとなる。

- (1)メインシステム、サブシステムを同時に、それぞれ独立して起動開始
- (2)サブシステムの起動完了時にメディアが装着されていれば再生を開始
- (3)メインシステムは起動完了時、サブシステムへ権限委譲開始が可能になった旨を通知

- (4)サブシステムからシステム状態情報を共有メモリへ転送し、操作要求信号の受付とサブシステム側 MW の制御を停止
- (5)メインシステム側では共有メモリからシステム状態情報を読み出し、操作要求信号とサブシステム側 MW の制御権を取得
- (6)簡易アプリを終了し、メインアプリでコンテンツの再生を継続

上記(4)~(6)の流れを図 3 に示す。

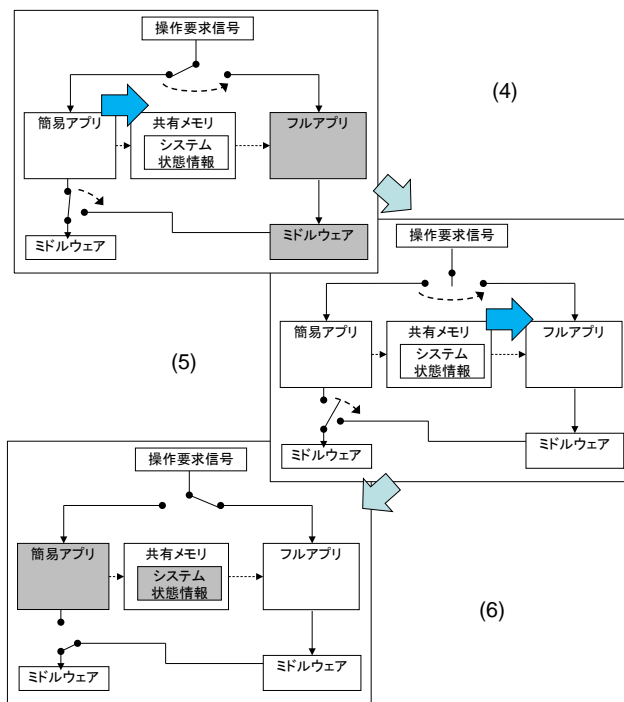


図 3. 権限委譲処理概要

3. 提案方式の適用検討

本章では、前章で説明したハイブリッド OS システムを車載 BD プレーヤーに適用する例について説明する。BD では、規格において UDF ファイルシステムドライバ、Java 仮想マシン (Java VM)、インターネット接続機能が必要になるため、提案方式の適用例として適する。

3.1 システム構成

提案方式の適用例となる車載 BD プレーヤーのシステム構成を図 4 に示す。このシステムは 2 つの CPU を持ち、2 つの CPU がそれぞれ CPU-S, CPU-M として動作する。各 CPU に対して、相互に通信するための CPU 間通信機構を持ち、また RAM 領域の一部は両方の CPU からアクセス可能な共有メモリ領域である。さらに、ユーザー操作要求信号の入力先は CPU-S, CPU-M の間でソフトウェアから切り替えが可能である。A/V デコーダ及び映像出力制御ブロックは CPU-S にのみ接続されている。

ソフトウェアは、CPU-S はサブシステムとして働き、RTOS 上で簡易アプリを動作させる。サブシステムはデコーダや映像出力制御ブロックの直接制御を行い、また DVD の再生システムとしても働く。CPU-M は汎用 OS 上でメインシステムを動作させるとともに、BD 再生システ

ムとしても働き、BD 再生に必要な Java VM の動作とネットワークデバイスの制御を行う。

また、この構成においては、サブシステム側に DVD の再生機能があるため、メインシステムの起動完了前にサブシステム側の簡易アプリで DVD の再生を開始することができる。

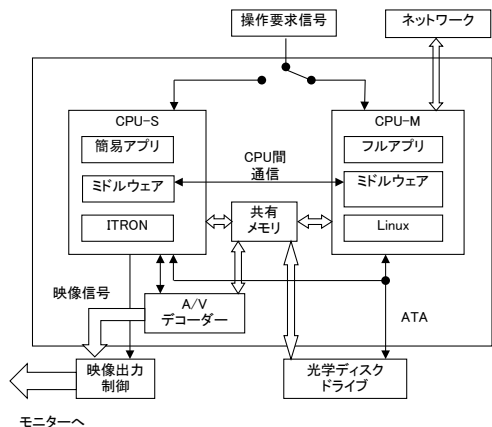


図 4. BD プレーヤーシステム

3.2 システム状態情報

簡易アプリからフルアプリへ引き継ぐべきシステム状態情報としては、以下のようなものがあげられる。

- ・再生状態情報：再生中のコンテンツ名、再生時間、再生速度等、GUI の表示に使用する情報である。権限委譲開始時点での最新情報を格納する。
- ・再生イベント通知：再生中に発生し、ユーザーへの通知や再生制御に使用される情報、例えば、再生状態の変化通知等である。権限委譲開始から、共有メモリ書き込みまでに発生した再生イベントについて、複数の場合には発生順も保ってメインアプリへ引き渡す必要がある。
- ・プレーヤー設定情報：周辺機器への設定データや、再生制御処理が使用するパラメータ系（音声ストリーム設定、字幕設定）等、サブシステムとメインシステムで共通に使用するプレーヤー設定情報についても、メインシステムが再生制御を開始する前に引き渡す必要がある、権限委譲のタイミングが適切と考えられる。

3.3 起動処理及び権限移譲処理

各 CPU 及びソフトウェアでの起動処理について説明する (図 5)。

3.3.1 電源 ON からの処理 (CPU-S)

CPU-S での起動時は、以下の処理を行う。

- (1) RTOS の起動
- (2) MW, I/O ドライバ, 簡易アプリ起動
- (3) 起動画面出画
- (4) 簡易アプリのみで再生可能なメディアである場合にはファイルシステムのマウントを行い、再生を開始

3.3.2 電源 ON からの処理 (CPU-M)

CPU-M での起動時は、以下の処理を行う。

- (1) 汎用 OS の起動
- (2) MW, I/O ドライバやライブラリのロード, ローカルストレージのマウント等
- (3) フルアプリの起動

3.3.3 権限委譲

簡易アプリ起動時点でフルアプリが起動していなければ、光学ディスクドライブやその他メディアは簡易アプリによって再生するが、フルアプリが起動した後は簡易アプリからフルアプリに再生制御の権限を移し、フルアプリで再生を行う必要がある。この権限委譲は以下の流れで行う。

- ① フルアプリ起動完了
フルアプリの起動完了後、フルアプリは簡易アプリに対して CPU 間通信によって権限委譲開始が可能である旨を通知する。
- ② 権限委譲開始
簡易アプリ側では、①の通知を受けて権限委譲が可能になったら、簡易アプリ GUI による描画を消去し、ユーザー操作の受付を停止する。
- ③ システム状態情報転送
簡易アプリは共有メモリにシステム状態情報を書き込んだ後、CPU 間通信を用いて書き込み完了をフルアプリへ通知する。続いてフルアプリではシステム状態情報を読み出し、簡易アプリへ取り込みの完了を通知する。
- ④ システム状態情報の取り込み
フルアプリでは読み出したシステム状態情報に含まれる再生制御情報に基づいてフルアプリ GUI を出画し、また再生イベント通知に基づいて再生制御を開始する。さらに、プレーヤー設定情報をメインシステムへ設定する。
- ⑤ 制御信号の入力先切り替え
サブシステム側 MW では③を受けて、制御主体をメインシステムへ切り替え、簡易アプリの動作を停止する。同時にフルアプリではユーザー操作信号の受付を開始し、権限委譲を完了する。

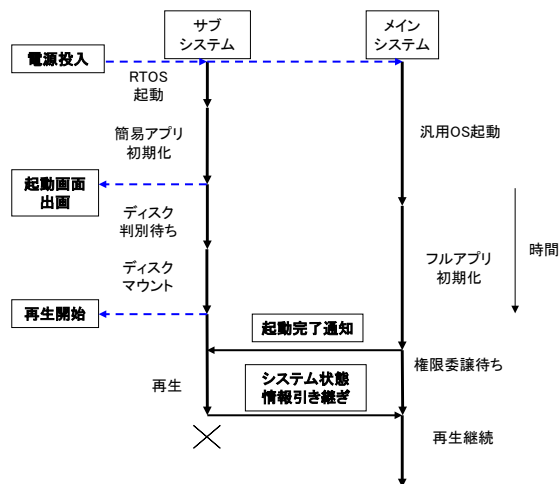


図 5. 起動処理の流れと権限委譲のタイミング

4. 提案方式による効果

提案方式の効果確認は、3章で述べた内容を実際に試作して行った。以下にその内容を記す。

4.1 測定内容

提案方式を適用した場合としない場合のそれぞれにおいて、起動画面出画までの時間と、コンテンツの再生開始までの時間を計測した。

起動画面の出画時間は、システムの電源を On してから起動画面出画までの時間を計測し、また、コンテンツの再生開始までの時間は、DVD ディスクをドライブに挿入した状態でシステムを起動し、DVD 出画までの時間を計測した。

測定はデジタルビデオカメラを用いて動作状況を外部から 30fps で撮影し、電源投入時点と出画時点の2つのフレームの間のフレーム数をカウントすることで行った。また、掲載した測定時間は5回測定の平均とした。

4.2 システム構成

システム LSI として、複数 CPU を内蔵する SoC (System On Chip) を採用した。この SoC は映像デコーダと、音声デコーダを内蔵する。また、外部 DRAM の一部を共有メモリとして使用し、各 CPU の間で割り込みを利用して CPU 間通信を行う。本システムでは、RTOS には μ ITRON、汎用 OS には Linux を使用した。

4.3 起動時間評価結果

4.3.1 起動画面出画までの時間計測結果

提案方式を適用前後の計測結果を図 6 に示す。本提案方式を適用前は起動画面の出画までに 8.9 秒かかっていたものが、提案手法を用いることで、2.6 秒までの短縮を実現した。Linux のみを用いたシステムでは達成困難な処理時間での起動画面出画を実現している。

4.3.2 DVD 出画までの時間計測結果

起動から DVD-Video の出画までの時間を、提案手法を用いることで、10.5 秒から 9.7 秒と約 1 秒の短縮を実現した。

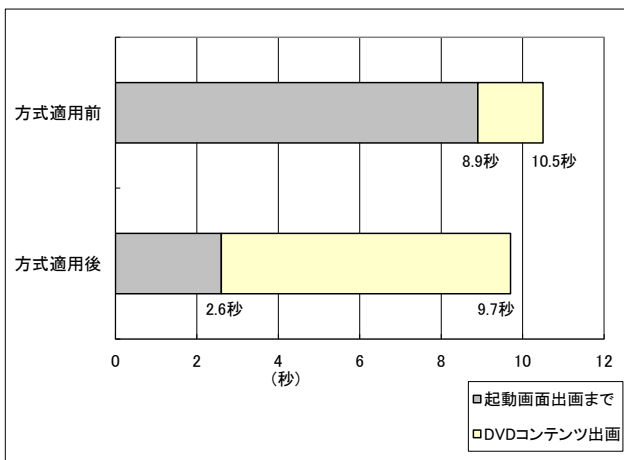


図 6. 出画時間の比較

4.4 考察

4.4.1 起動画面出画時間について

起動画面の出画時間に関しては、提案方式では、起動画面の出画処理は μ ITRON のみで行っており、Linux の起動を待たずに行っている。Linux では画面出力が可能になるまでに 10 秒程度かかっており、本方式による高速化の効果が確認された。

4.4.2 DVD 出画時間について

一方、DVD の出画時間については、提案方式における出画時間短縮が 1 秒弱に留まっている。この理由としては、全体の処理時間の中で、光学ディスクからのデータ読み出しが可能になるまでの時間がボトルネックになっているためであると考えられる。

図 5 に示したように、簡易アプリでは光学ドライブがディスク判別処理を終了するまでの間の待ち時間が生じていることが分かる。従って、光学ドライブのディスク判別処理を高速化することで、出画時間がさらに高速化することが可能であると考えられる。

5. まとめ

本報告では、ハイブリッド OS システムを用いた車載メディアプレーヤーの起動高速化方式を提案した。提案方式では、簡易アプリが持つ制御権とシステム状態情報をメインアプリの起動後に引き渡す。このことにより、汎用 OS のライブラリ群を有効に活用できるとともに、RTOS の起動高速性を利用することが可能となる。また、提案方式を車載 BD プレーヤーの試作機に適用した結果、6.3 秒の起動高速化を実現した。

一方、提案方式では、権限委譲処理により起動処理が複雑化しており、デバッグが困難になるという問題点もある。この点は今後の研究課題としたい。

参考文献

- [1] 中島 達夫. “組み込みオペレーティングシステム概論”, 映像情報メディア学会誌, Vol.63, No.5, pp.633~637 (2009).
- [2] 田丸 喜一郎. “5.組み込みプラットフォームの動向”, 情報処理, 45(7), 699-703 (2004-07-15).
- [3] 木内 志朗. “Linux を使った組み込みシステム開発”, Interface 増刊 TECH I. Vol.16 「組み込み Linux 入門」第 1 章.(2003)
- [4] 遠藤 幸典, 菅井 尚人, 山口 義一, 近藤 弘郁. “シングルチップマルチプロセッサ上のハイブリッド OS 環境の実現—システムアーキテクチャー”, 情報処理学会全国大会講演論文集, 66th, 1.9-1.10(2004).
- [5] 遠藤 幸典, 菅井 尚人, 山口 義一, 近藤 弘郁. “シングルチップマルチプロセッサ上のハイブリッド OS 環境の実現—OS 間インタフェースの実装—”, 情報処理学会全国大会講演論文集, 66th, 1.11-1.12 (2004).
- [6] 藤田 嘉和, 小野 俊幸, 山口 達也, 堀山 実. “車載用 Blu-ray デッキメカニズムの開発”, 富士通テン技報. Vol.30, No.1.(2012)