

組込みシステム集約基盤「誉」におけるリアルタイム OS 集約とコスト評価

奥川 莞多[†] 金城 聖[†] 毛利 公一[†]

[†] 立命館大学情報理工学部

1 はじめに

組込みシステムで高性能化や多機能化が進んでおり、特に車載システムのような大規模なシステムでは OS を使用しないソフトウェアや RTOS だけでなく、汎用 OS といった多様なソフトウェア (まとめてシステム制御ソフト) が使用される。それに伴い、使用機器の増加によるハードウェア構成の複雑化や内部スペースの圧迫などの問題がある。一方で、組込み機器に搭載するプロセッサのマルチコア化が進んでいるが、既存の RTOS や OS を使用しないソフトウェアはマルチコアプロセッサを十分に活かしてきていないという現状がある。これらの問題は、これまで個別の機器で動作していたシステム制御ソフトを 1 つの機器に集約することで解決が可能である。

これらの背景から、我々は、マルチコアプロセッサを搭載した 1 つの機器に既存のシステム制御ソフトを容易に集約できる軽量な基盤として誉を開発してきた [1]。本論文では、組込みシステム集約基盤「誉」において、TOPPERS/ASP3[2] と FreeRTOS[3] を対象として集約を実現し、その際の集約コストを評価したので報告する。

2 組込みシステム集約基盤「誉」

2.1 概要

各システム制御ソフトにリソースを割り当て、容易なシステム集約を実現する基盤として誉を開発してきた。誉は組込みシステムでの利用や既存のシステム制御ソフトの集約を想定しており、リアルタイム性や移植性などの要件を満たす必要がある。そこで、誉は図 1 のように LPAR 方式と準仮想化を組み合わせたアーキテクチャで、組込みシステムの集約を実現する。基本的には LPAR 方式を用いて、システム制御ソフトに CPU やメモリなどの専有リソースを割り当て、直接操作させることでリアルタイム性を実現している。また、LPAR では実現が困難な処理に関しては準仮想化を用いて、誉が準仮想化インタフェース (誉 API) を提供することで、誉がシステム制御ソフトの処理を代行する形でデバイスの共有などの機能を実現している。

2.2 誉が提供するインタフェース

誉はコア抽象化やコア依存処理代行、デバイス共同利用などの機能をゲスト OS (誉上で動作するシステム制御

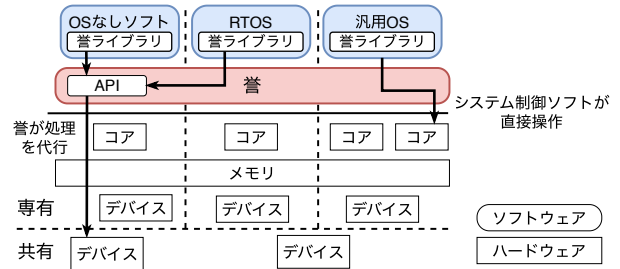


図 1 誉によるシステム集約のアーキテクチャ

表 1 誉 API の一覧

API 名	用途
homare_get_sysprms	コア抽象化機能初期設定
homare_putChar	UART 出力
homare_irq_config	単一の割り込み操作依頼
homare_irq_config_multi	複数の割り込み操作依頼
homare_get_cpuid	コアディスクリプタ取得
homare_cd_to_cn	物理コア番号取得

ソフト) に表 1 に示す誉 API によって提供している。

集約時にゲスト OS はどのコアでも動作する必要があるが、特定のコアで動作することを想定されているものがあり、複数のシステム制御ソフトを集約する際の課題となる。そこで、homare_get_cpuid により抽象化した物理コア番号 (コアディスクリプタ/cd) を提供し、ゲスト OS は cd を使用することで、特定のコアへの依存を解消している。また、ゲスト OS のコア依存処理を誉 API に置き換え、誉が処理を代行することで、コアへの依存を解消できる。homare_irq_config は、依頼する操作と cd を引数に指定し、各コアに依存する割り込み有効/無効設定や割り込み要因取得などの処理を誉に依頼できる。これらの誉 API により、ゲスト OS への柔軟なコア割り当てを可能にしつつ、システム集約を実現している。また、UART といった誉やゲスト OS 間で共有したいデバイスの処理は homare_putChar 等によって、デバイスの共同利用を実現している。

これらの誉 API を適用する際に、ゲスト OS に誉 API のスタブとしてライブラリ (誉ライブラリ) を提供することでゲスト OS の移植性を確保している。

3 リアルタイム OS 集約時の改変と動作確認

今回は、TOPPERS/ASP3(ASP3) と FreeRTOS を対象に、誉上で動作させる改変をし、動作を確認した。

Real-time OS Integration and Cost Evaluation in Embedded System Integration Platform “Homare”

Kanta Okugawa[†], Akira Kanashiro[†], and Koichi Mouri[†]

[†]College of Science and Engineering, Ritsumeikan Univ.

```
// TOPPERS/ASP3 のコード
sil_rwr_mem(BCM283X_COTIMER_INTCTL, sil_rew_mem(
    BCM283X_COTIMER_INTCTL) & ~(1U<<3));
// FreeRTOS のコード
*COREO_TIMER_IRQCNTL |= (1U << 3);
// 修正後
homare_irq_config(0,99,IRQOP_ENABLE_INTERRUPT);
```

図 2 コア依存のタイマ割り込み許可処理の置き換え例

3.1 リアルタイム OS の改変

誉では、ゲスト OS に対して以下の改変が必要となる。

- LPAR 方式におけるリソース競合を避ける改変
- 準仮想化における誉 API を適用する改変

LPAR 方式の改変では、誉やゲスト OS 間で、メモリやタイマ、UART などのリソース競合を避ける必要がある。メモリリソースは、リンカスクリプト等を修正することで、ゲスト OS 間で重複しないように改変する。また、Generic Timer というコアごとに割り込みを発生させるタイマを使用するように改変し、ゲスト OS にタイマを専有させる。さらに、UART や LED などのデバイスに関しても使用しないか競合しないように改変する。

準仮想化の改変では、2.2 節で述べたようなコアに依存する処理や共有したいデバイスの処理を誉に処理代行を依頼する API に置き換える必要がある。改変内容は、誉ライブラリの追加と誉 API の適用である。

まず、誉 API を利用するために、ゲスト OS のイメージに誉ライブラリを含める。つぎに、コア抽象化機能の初期設定として起動処理に `homare_get_sysprms` を追加する。また、コアに依存する処理や共有するデバイスを利用する処理は誉 API に置き換える。今回の対象 OS では、図 2 にのようにコア 0 に依存するタイマ割り込みの処理を誉に代行することで、コアに依存する処理を解消する。また、誉とゲスト OS 間で共同利用するデバイスの処理は、誉に処理代行を依頼する API を使用するように改変する。

3.2 動作確認

誉はメニーコアを想定しているが、今回は 4 つのコアが搭載されている Raspberry Pi 3 Model B を使用した。1 つのコアは誉が使用し、ASP3 と FreeRTOS は 3 つのコアの内どのコアにも割り当て可能である。また、それぞれ複数での並行動作を確認した。

4 改変コスト評価

システム制御ソフトの移植性を確保する上で、誉 API が有用であるかを示すために、表 2,3 の ASP3 と FreeRTOS の集約時の改変コストをもとに評価した。

表 2 では、ビルドファイルを含めた全体の行数、ファイル数と誉 API を適用する際の改変 (追加・削除) を比較している。全体と比較し、修正するファイル数は少ないことがわかる。

表 2 誉 API 適用時の改変コストと全体の比較

対象 OS	行数		ファイル数	
	全体	修正量	全体	修正数
ASP3	77153 行	186 行	515 個	10 個
FreeRTOS	29451 行	127 行	181 個	6 個

表 3 誉 API に対応するための改変コスト

対象 OS	誉 API 適用		誉ライブラリ追加
	箇所	修正量	修正量
ASP3	11 箇所	34 行	152 行
FreeRTOS	4 箇所	7 行	120 行

表 3 は誉 API を適用する際の改変の内訳である。誉 API の適用は、ASP3 で 34 行、FreeRTOS で 7 行の改変という結果となり、十分小さい改変コストである。また、OS の構成自体を変更する修正はなく、コードの置き換えと追加のみで、置き換えの際は図 2 のように誉 API を利用するため、容易である。適用箇所は、ゲスト OS の割り込み処理と起動処理などであり、対象の処理を理解すれば、検討は困難ではない。誉ライブラリ追加の改変は、全体の改変行数のうち FreeRTOS では 120 行、ASP3 では 152 行であったが、3 つのファイルの追加と Makefile 等のビルド設定ファイルの編集であり、改変コストは十分小さい。ASP3 のようにスタック準備前に誉 API を使用する場合も、あらかじめ準備されたライブラリのコードを追加するのみで利用可能である。

5 おわりに

組込みシステムの容易な集約を実現する基盤である誉において、TOPPERS/ASP3 と FreeRTOS の集約を実現した。また、集約時の改変コストを評価し、誉 API 適用においてはライブラリの追加と数十行の改変であり、容易に集約可能であった。今後は、Linux 等の汎用 OS の集約を実現し、リアルタイム性や移植性を評価する。

参考文献

- [1] 林海豊, 井内 晴菜, 毛利 公一: 組込みシステム集約のためのマルチコア制御基盤「誉」におけるコアの抽象化, 研究報告システムソフトウェアとオペレーティング・システム (OS), Vol. 2022-OS-154, No. 9, pp.1-8(2022)
- [2] TOPPERS Project, Inc: TOPPERS プロジェクト /ASP3, TOPPERS プロジェクト (オンライン), 入手先 <https://www.toppers.jp/asp3-kernel.html> (参照 2023-1-11)
- [3] Amazon Web Services, Inc: FreeRTOS - Market leading RTOS(Real Time Operating System) for embedded systems with Internet of Things extensions, 入手先 <https://www.freertos.org> (参照 2023-1-11)