3A-3

組込み機器における マルチコアプロセッサを用いた高可用性システムの一提案

佐藤　龍一†，平田　明†，虻川　雅浩†

三菱電機(株)　情報技術総合研究所††

1. はじめに

組込みシステムは CPU 処理性能の向上に伴い多様な機能が要求され、ソフトウェア規模が増大すると共に構造が複雑になっている。このため、ソフトウェアの不具合を完全に無くした状態で製品出荷することが難しくなっており、対策が急務となっている。また、サードパーティアプリをインストールして使用するケースも増加し予期しない障害発生により、システムが継続して動作できない状況に陥る場合が考えられる。

本提案ではソフトウェア障害による、予期しないシステムの再起動などによるダウンタイムを小さくし、エンドユーザにストレスを与えない、高可用性システムの方式について示すものである。

2. 課題

システムの可用性を高め連続運用を可能にする方法として、システム全体を多重化し、障害発生時にフェールオーバーさせる手法がある。しかしこの手法を組込みシステムに利用するには以下の問題がある。

① システムの規模が大きくなる
② 部品点数が増え、コスト高になる
③ 消費電力が大きくなる

上記問題を考慮し、システム全体を多重化することなく、組込みシステムにおいてソフトウェア障害に対する高可用性を実現する必要がある。

3. マルチコアを用いた高可用性システム

3.1　概要

マルチコアプロセッサが搭載されたシステムで CPU コアを通常の動作に使用する「主系」と、異常発生時に切換えて使用する「待機系」に分割し、マルチコア内部に２重系システムを構築する。障害発生時に待機系へ即座にシステムを切換えて動作させることで、ダウンタイムを短くすることを可能にする。

Proposal of high availability systems using multi-core processor.

†Ryuichi Sato, Akira Hirata, Masahiro Abukawa
‡Information Technology R&D Center, Mitsubishi Electric Corporation

3.2　２重系システムの構築

Dual コア搭載のシステムを例にマルチコアを用いた２重系の構築について説明する(図１)。

マルチコアの CPU コアを利用し主系システムと待機系システムに分け、それぞれ OS を搭載し、主記憶装置(以下メモリ)を論理的に分割して使用し、各系で独立した領域にアクセスする。

なお、本提案では既存のマルチコアシステムを流用することを想定し、仮想化技術を利用しない方法で検討を行った。このため、主系/待機系は排他的に I/O を利用するように制御する。I/O へのアクセスは主系のみアクセス可能とし、待機系からは起動時以外はアクセスを行わない制御を行い、アクセス競合が発生しないようにする。

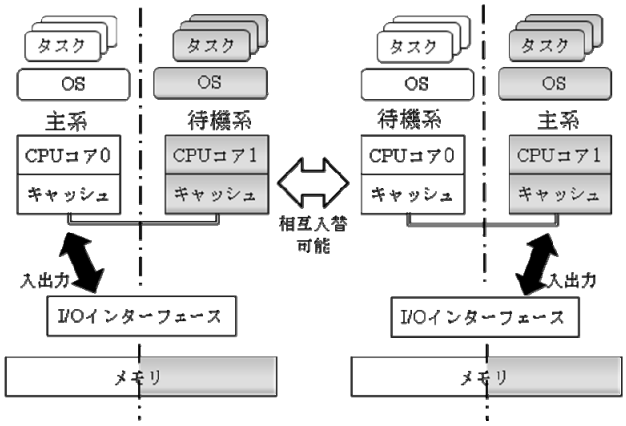

図１　Dual コアにおける２重系システムの構築

3.3　再起動時間短縮による高可用性の実現

(1) ２重系システムの起動

システム起動により、主系、待機系ともに起動し、各系で BOOT、OS の起動、ソフトウェアの初期化処理を行う。主系ではハードウェアの初期化を行い、ハードウェアを制御し、システムとしての通常動作を開始する。待機系はハードウェアの初期化は行わず、待機状態に入り主系からのイベントの通知を待つ。待機系は CPU の状態を SLEEP 状態に遷移させ、低消費電力化と高速な復帰を可能にする。待機系は主系からのイベント通知、または自身の周期タイマにより SLEEP から復帰し主系稼働状態の監視を行う(図２)。また起動シーケンスを高速化するために起動完了イメージをメモリへ展開する技術と組み合わせると、高速に起動し待機状態へ遷移するまでの時間を短縮することができ、さらに可用性を向上させることができる。

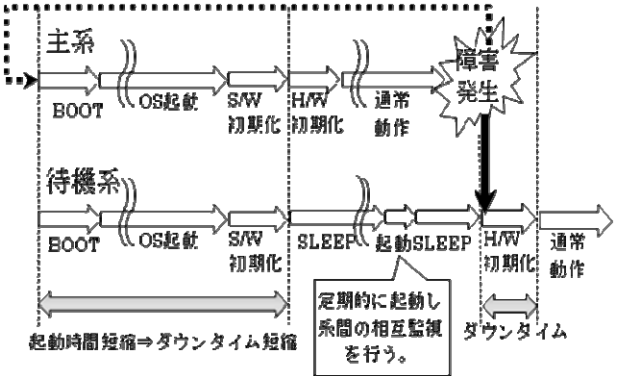

図２　マルチコアを用いた２重系システム動作

(2) 異常の検出

チェックサムを用いたメモリ内容の異常チェックや、例外などの CPU エラー情報による主系内部の異常検出機能に加え、主系と待機系間で状態の相互監視を行い、各系の OS の状態を含めた異常検出を可能にする。系間の相互監視の具体例を示す。CPU 間通信と共有メモリを使用し、監視される系で周期的に共有メモリの内容を更新し、監視する系へ更新完了を通知する。監視する系では CPU 間通信を受信し共有メモリ内容が期待する内容かどうかをチェックし主系が正常に稼働しているか判断する。

異常の自己検出に加え、別システムで動作する待機系から相互監視することでソフトウェア要因による障害の早期検出と検出範囲の拡大を可能にする。

(3) 主系から待機系への遷移

主系で動作継続不可能な異常を検出した場合は、CPU 間通信を使用し「系切換通知」を待機系へ通知する。その後、主系は再起動し、新たに待機系として起動、ハードウェア初期化は行わず待機状態へ遷移する。「系切換通知」を受信した待機系は、新たに主系として動作を開始しハードウェアの初期化を行い、通常動作へ遷移する。

待機系で主系の異常を検出した場合は待機系から主系へ CPU 間通信で「系切換要求」を通知し系の切換を要求する。「系切換要求」を受信した主系では「系切換通知」を待機系へ通知し、主系で障害を検出した時と同様に、主系は再起動し新たに待機系として起動、待機系は新たに主系として動作する。

異常を検出後、新主系ではハードウェアの初期化を行い通常動作へ遷移する。この時、検出した異常を解析し CPU 間通信と共有メモリを使用してエラー要因を新主系へ通知する。新主系のハードウェア初期化処理で、エラー要因と関係のない正常に動作していたハードウェアの I/O 情報はそのまま利用し、初期化が必要なハードウェアのみ初期化することで処理時間を短縮し系切換え時間を短くする。

(4) CPU コア数の不均衡分割

４コア構成のマルチコアにおいて２重系を構成する例として、主系に３個、待機系に１個を割り当てた場合を示す(図３)。

コア３個の主系ではシステムの全機能をサポートする。コア１個の待機系ではシステムを動作させるために必要最低限の機能をサポートする。主系で動作継続が不可能な障害発生時、待機系でシステムとして最低限のサービスを提供しつつ、主系を再起動する。再起動完了後、主系へ再度遷移させ全機能のサポートを行う。最低限のリソース上でバックアップシステムを構築し可用性を向上させることができる。

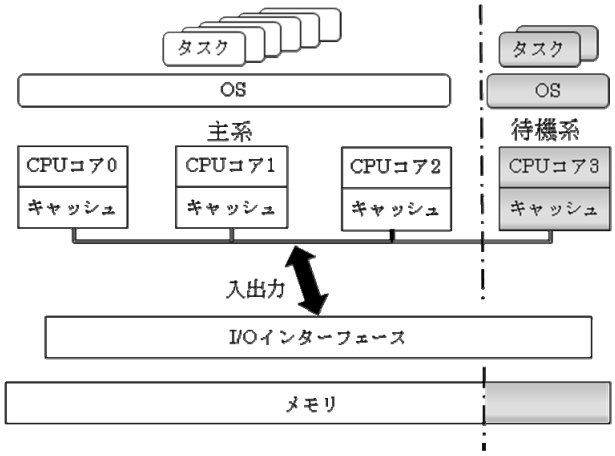

図３コア数不均衡分割による２重系構築

4. 課題

障害の発生状況に応じて必要なハードウェアの初期化のみ実施し処理時間を短縮する案について、実現可能な方法の検討と効果の確認が必要である。エラー要因から、影響範囲を特定するにはソフトウェア構成、ハードウェア構成、すなわちシステムの構成に依存した部分を解決する必要があり、一意に手法を決定することができない。ハードウェアの仕様で特定のタイミングにおける操作が必要な場合や、システムによってはエラー要因を解析するよりも、全てのハードウェアの初期化を行った方が高速に復帰できる可能性もある。特定のシステムにおいてモデリングを行い手段と効果を検討する必要がある。

5. おわりに

今後は特定のシステムに本手法を適用し、2 重系システムの構築と上記課題を検討し、効果を確認していく予定である。

参考文献

[1] 向殿政男他，“コンピュータシステムの高信頼化技術入門”，日本規格協会