2A-04

組込みLinuxにおける障害対処機能の実現方式検討

堀井　圭祐†　　飯田　博之†　　落合　真一†

三菱電機株式会社　情報技術総合研究所†

1. はじめに

近年、産業分野などの組込みシステムにおいても Linux の適用が拡大している。産業用途向けの組込みシステムでは高い信頼性が要求されており、障害発生時の早急な原因究明と適切な対処が必要である。しかし、Linux が標準的に備えている障害対処機能では不十分であり、システム要件を満たすために機能を追加する必要がある。これまで、このような障害対処機能を実現するためにはカーネル内部に直接手を加えることが一般的であった。しかし、従来の方式ではカーネルの内部構造が変更される度に、メンテナンスが必要でありコストの増大が課題である。

そこで、本稿では Linux カーネルの外部に障害対処機能を追加することによって、カーネル内部には極力手を加えずに実現する方式について検討した。

2. 従来方式の課題

従来、Linux において障害対処機能を追加するためにはカーネル内部の例外ハンドラやパニック関数などの障害検出機構に対して、修正・変更を行い独自機能として追加する必要があった。しかし、カーネル内部に手を加える場合、メンテナンスコストの増大が課題である。

Linux ではカーネルをアップデートすることによって、セキュリティパッチの適用や最新デバイスへの対応などのメリットを得ることが可能である。特にセキュリティパッチについては、ディストリビューションから長期的に提供されるものであり信頼性の確保・長期保守の実現といった観点からも重要なサービスである。しかし、カーネルをアップデートすると内部構成が変更される可能性があり、カーネル内部に手を加えていた場合、カーネルの変更に応じてメンテナンスする必要がある。具体的には以下のようなメンテナンスコストが発生すると考えられる。

(1) 調査コスト
　カーネルのアップデートにより変更された内容をソースコードレベルで調査し、独自機能に対する影響を明確にする必要がある。
(2) 修正コスト
　カーネルの変更に応じて、独自機能を修正する必要がある。
(3) 試験コスト
　修正した独自機能が問題なく動作することを確認する試験を行う必要がある。

また、カーネルのソースコード規模が増大するにつれ、独自機能に対する影響範囲も拡大する恐れがある。このような Linux のカーネルに対しては、極力手を加えないようにすることが望ましい。

3. Linuxでの障害対処機能実現方式

3.1. 構成

カーネル内部へ手を加えずに障害対処機能を実現する方式として、Linux と障害対処用 OS の 2OS 構成を検討する。これまでにも 2OS 構成により OS の資源負荷状況を監視する方式が検討されてきた[1]。しかし、従来の方式では OS 内部の障害が発生したことを検知することは困難であった。そこで、本方式は図 1に示すように Linux 自身で障害発生を検出し、障害対処用 OS にて障害対処を実施するものである。Linux では障害検出後に、障害の発生要因を特定するための障害情報の収集、および障害対処用 OS 上の障害対処モジュールの呼び出しを行う。障害情報とは CPU レジスタの情報や障害発生時に動作していたプロセスの情報などであり、外部から収集することは困難である。そのため、障害情報の収集は Linux 側で行うも

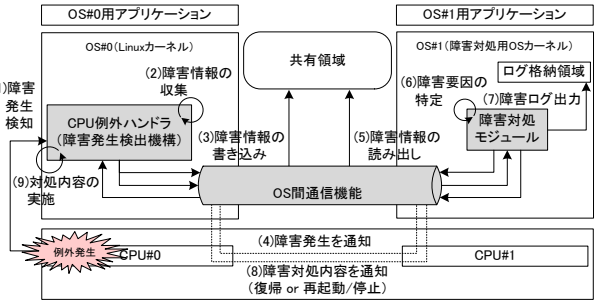

図 1　機能構成と処理手順（CPU 例外発生）

An examination of failure detection methods for the embedded linux

Keisuke Horii, Hiroyuki Iida, Shinichi Ochiai

† Information Technology R&D Center, Mitsubishi Electric Corporation

のとする。障害対処用 OS では Linux からの障害発生通知を受け、障害原因を究明し対処内容を定義・実行する。

図 1より、本方式は主に以下の 3 つの要素から構成される。

- OS 間通信機能
  2OS の協調動作を実現するための機能である。OS 間での障害情報の送受信を実現するために、共有領域に対する書き込み・読み出し機能を提供する。また、Linux から障害対処用 OS への障害発生通知、および障害対処用 OS から Linux への障害対処内容通知を実現するためのプロセッサ間割り込み発行機能を提供する。
- 障害発生検出機構
  Linux 内で障害が発生したことを検出する機構である。本方式により検出する障害内容、および検出機構を表 1に示す。なお、アプリケーション監視部とはアプリケーションに送付されたシグナルをキャッチし異常終了されたことを検出するための独自機構である。アプリケーション監視部は、ユーザ空間で動作するためカーネル内部に手を加えずに実現可能である。障害発生検出機構ではデフォルトで定義されている障害対処処理は実行せず、収集した障害情報を共有領域へ書き込みを行い、障害対処用 OS に障害が発生したことを通知する。
- 障害対処モジュール
  障害への対処を実現するモジュールである。Linux からの障害発生通知を受けると、共有領域から障害情報を取得する。取得した障害情報に基づいて発生要因を特定し、障害対処を実施する。障害対処内容としては障害ログの出力と、OS の処理が継続可能であるかの判定を行う。判定の結果、決定した対処内容を Linux へ通知する。また、本モジュールでは障害対処用 OS 内で発生した障害についても対処できるようインタフェースを備えるものとする。

表 1　Linux で検出する異常分類

障害分類	内容	検出機構
S/W 障害	カーネル内エラー	カーネルパニック処理
S/W 障害	アプリケーションエラー	アプリケーション監視部
S/W 障害	CPU 例外	CPU 例外ハンドラ
H/W 障害	NMI 通知	NMI ハンドラ

3.2. 効果

本方式により Linux の障害対処機能を追加することによって、以下のような効果が得られると考えられる。

(1) カーネル内部への修正・変更を低減
Linux のカーネル内部への修正・変更内容としては、障害情報を収集する機能と障害発生を通知する機能の実装のみとなる。また、Linux はカーネル内部に様々な障害検出機構を備えているため、新規に検出機構を追加する必要はない。

(2) 障害を統一的に管理することが可能
Linux の各障害検出機構から共通のモジュールを呼び出すことによって、障害の発生を漏れなく把握することが可能であり、異なる検出機構で検出した障害であっても統一されたフォーマットによるログを出力することが可能である。また、OS 処理の継続・中止を共通のモジュールで判断するため、システムの一貫性を保持することが可能である。

(3) 2OS による信頼性向上
産業用途向けの組込みシステムでは一定の周期毎に処理を実行する必要があり、処理の継続が可能な障害発生時には周期処理を継続することが望ましい。しかし、1 つの OS 上で障害対処を行う場合、障害対処機能により周期処理の実行が阻害され、遅延が生じる可能性がある。本方式は 2OS 構成であるため、Linux 上でアプリケーション障害などの軽度の障害が発生した場合でも、周期処理を満たしつつ障害対処用 OS 側で障害のログを記録することが可能である。

4. おわりに

本稿では、Linux のカーネルに極力手を加えずに障害対処機能を実現する方式として、Linux とは別の障害対処用 OS 内に障害対処モジュールを追加する方式を検討した。本方式によりカーネ内部への変更・修正を低減しつつ、Linux の各障害検出機構から共通のモジュールを呼び出すことで障害を統一的に管理することが可能である。

2OS 構成とする場合、OS 間通信がオーバヘッドになる可能性があるため、今後は本方式により性能目標を達成可能であるか評価する予定である。

参考文献

[1] 岡部、村山、攝津、「ハイブリッド OS を用いた OS 監視方式の提案」、電子情報通信学会 総合大会 2009 年 3 月