カーネル拡張モジュール機能を用いたOSデバッグの実現

中村　哲人†　　畑崎　恵介†　　芹沢　一‡
（株）日立製作所　中央研究所†
（株）日立製作所　システム開発研究所‡

1．はじめに

近年急速に注目を集めるオープンソースOSであるLinuxは、コミュニティによって仕様決定や機能保守が行われており、さらに仕様変更や追加が頻繁に行われることにより、品質の安定化が重要な問題事項となっている。Linuxのさらなる適用範囲拡大のためには、この品質向上が重要であり、それを支援する様々なデバッガが各種提案されている。

我々はこの問題に対して、Linuxが持つカーネル拡張モジュール機構を利用したOSのデバッグ方法を提案する。カーネル内のイベント発生時に駆動されるデバッグ機能を動的に変更可能にすることにより、カーネル内で発生した障害解析に最適なデバッグ方式を、システムリブートなしにカーネルに組み込むことが可能である。

本稿では、Linuxカーネルのトレーサシステムである LKST(Linux Kernel State Tracer)上に、カーネルの拡張モジュール機能を用いて本方式を実装し、その有用性を確認した結果について報告する。

2．Linux Kernel State Tracer(LKST)の概要

LKSTはLinuxカーネルの状態変化をトレースし、障害発生原因の迅速な追求を支援する機能である。本機能は、日立、IBM、富士通、NECの４社で進めているエンタープライズLinux機能強化に向けた提案活動の一環として、富士通、日立を中心に開発を進めている。

図１に示すように、 LKSTは、あらかじめLinuxカーネルの各種イベント処理部分にフック関数を呼び出すようフックを挿入してある。そして、イベント発生時にはフック関数を呼び出し、フック関数がLKSTのイベントバッファへと記録する。イベントバッファは一杯まで記録されると、再び先頭から記録を開始する。

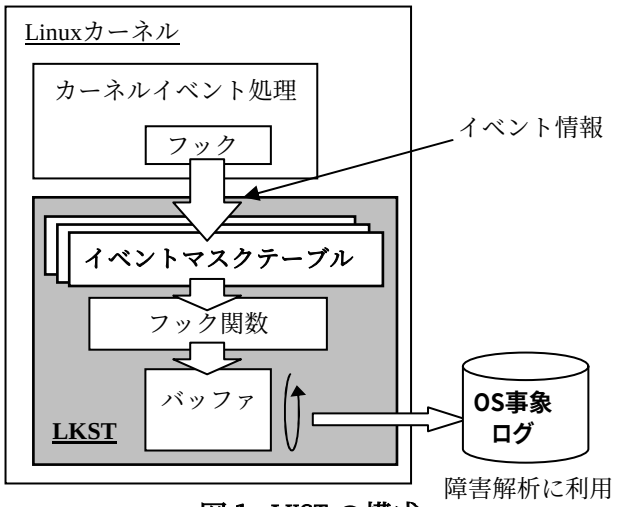

図１ LKSTの構成

カーネル内のフックはプロセスコンテキストスイッチやシステムコールなど、様々なイベント発生個所に挿入されている。また、記録されるイベントはイベントマスクテーブルによって選択可能である。

3．拡張モジュール機能を利用したLKST拡張

LKSTは通常イベントトレーサとして用いられ、その場合カーネル内のフックから呼ばれるフック関数はバッファにイベントを記録する役割を持つモジュールが用いられている。しかしながら、フック関数を、カーネル拡張モジュール機能を用いて他のデバッグ機能を実装したモジュールに置き換えることによって、様々なデバッグ方式を実装可能とした。図２に概要を示す。

この場合、LKSTを単なるフックインターフェースとして用いており、dprobesやLinux Security Module(LSM)のような他のLinuxフックインターフェースを用いても構わない。今回、LKSTを選択した理由を以下に示す。

(1) フック部分をアセンブラ記述で最適化済であり、高速である。
(2) LSMはフック挿入箇所がセキュリティ向けであり、汎用的でない。

Operating System Debugging with Using Kernel Extension Module
†Tetsuhito Nakamura、 Keisuke Hatasaki、 Hitachi Ltd. Central Research Laboratory
‡Kazuyoshi Serizawa、 Hitachi Ltd. Systems Development Laboratory

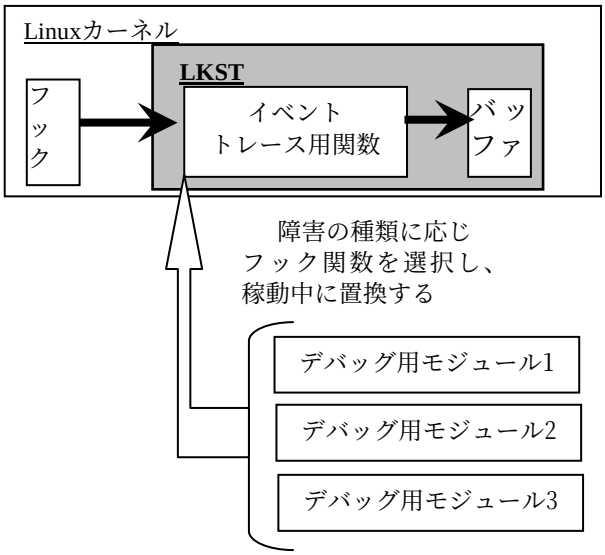

図 2 拡張モジュールを利用した LKST 拡張

4．OS デバッグの実例

本方式を用いた OS デバッグの一例を次に示す。

4.1 不正割込みの頻発

(1) 現象と原因

ある Linux カーネルにおいて、/dev/zero 仮想デバイスから通常のファイルへデータ転送を行うと、転送サイズの増加とともにシステム負荷が急上昇する。ディスク入出力割込みの頻発が原因であった。

(2) 解析方法

フックを通過した回数をカウントする機能を持つデバッグ用モジュールを準備し、上記現象が発生するカーネルと発生しないカーネル上で、評価を行った。その結果、上記現象の発生するカーネルにおいては、ディスク入出力割込みが非常に多く発生しており、さらなる詳細調査の結果デバイスドライバのバグを発見した。

(3) 利点

デバッグプリントを利用することも可能だが、この場合システムリブートが必要である。本方式であればリブートは不要であり、顧客の運用中システムを停止することなくシステムのデバッグが可能である。

4.2 カーネル内ページフォルトの原因分析

(1) 現象と原因

ページフォルトハンドラ評価のために、ハンドラに評価用コードを追加したカーネルにおいて、まれに OS がハングアップする。

i386 版 Linux では、カーネル用の仮想領域の一部はプロセスごとに独立したページディレクトリのエントリ(pde)を使ってマップされている(メモリマップの対応関係自体は共通)。このため、カーネル用仮想領域の確保の前に生成されたプロセスは、該仮想領域用の pde を持たない。この pde のコピーはマスタからのページフォルトを契機に行う。そのため、前述の仮想領域をページフォルトのハンドラで使用する場合は、ページフォルトがループするためカーネルが止まる。

(2) 解析方法

まずは、LKST によるトレース結果から、ハングアップ直前にページフォルトが多数発生していることを確認した。そこで、ページフォルトがループした場合は例外処理用のバッファに、原因となったリニアアドレスと pde、ページテーブルの内容等を記録するデバッグ用モジュールを作成し、上記現象を評価した結果。上記原因を突き止めた。

(3) 利点

ページフォルトハンドラ内にデバッグプリントを挿入する方法は、さらなるページフォルトを起こす可能性があり不可能。本方式では、このような例外ハンドラの問題においても、通常の動作に出来るだけ影響を与えないデバッグが可能である。

5．まとめ

本稿では、Linux が持つカーネル拡張モジュール機構を利用した OS のデバッグ方式を提案した。このデバッグ方式では、カーネル内のイベント発生時に駆動されるデバッグ機能を動的に変更可能にすることにより、カーネル内で発生した障害の解析に最適なデバッグ方式を、システムリブートなしにカーネルに組み込むことを特徴とする。

また、このデバッグ方式を用いて 2 件の OS バグについてのデバッグを行い。本方式の有用性を確認した。

6．参考文献

[1] 畑崎恵介、中村哲人、芹沢一：”システムの挙動に対応して動作の切り替えが可能なイベントトレーサ LKST の開発”、FIT2002、B-45

[2] 佐藤元信、高野了成、早川栄一、高橋延匡：”OS/omicron V4 におけるデバッグ支援環境の設計”、情報処理学会研究報告 システムソフトウェアとオペレーティングシステム No.083-004、1999

[3] Daniel P.Bovet、 Marco Cesati ”Understanding the Linux Kernel” O’REILLY

[4] Dprobes: “http://oss.software.ibm.com/developer/opensource/linux/projects/dprobes/”

[5] Linux Security Module：http://lsm.immunix.org