特集

急速に成熟するLinuxOS（機能・信頼性）

Rapid improvements of LinuxOS ー Features and Reliability ー

あらまし

LinuxはLinuxコミュニティと呼ばれるメーリングリストにおいて，ボランティアによる開発や使用実績作り，障害修正作成などの活動により，長足の進歩を続けている。現在のLinuxの最新カーネル版数である2.6は，メモリ管理機能やスケジューラ機能の改善により，さらに進歩したOSとなっている。

本稿では，富士通が基幹システムに必須と考え，コミュニティの一員として開発に参加したdiskdump（カーネルクラッシュ時やハング時の確実なダンプ採取），MCA強化（マシンチェック機構を活用した影響範囲の限定と回復），udev（デバイスの保守や増設などの前後で名称が変わらないデバイス名称の柔軟な決定機構），ホットプラグ（CPU，メモリ，IOバスなどの活性保守・キャパシティ変更機能）の四つの機能について述べる。

Abstract

Rapid improvements continue to be made in the ongoing development and usage of Linux through a mailing list of volunteers known as the Linux community. These volunteers contribute source code, report information on any trouble to the community, and quickly provide required bug fixes. The latest kernel version (2.6) reflects the many improvements made to Linux, primarily in memory management and the process scheduler. Consequently, Linux has become a much more advanced OS. This paper describes the four features being developed by Fujitsu (in conjunction with the Linux community) for mission-critical systems. These include the 1) reliable crash dump feature (diskdump) used in case of a Kernel crash, for example, 2) enhanced MCA to minimize the effects of hardware failure and recover from failure by using MCA feature, 3) persistent device naming feature (udev) that prevents device names from being changed before and after maintenance and expansion, and 4) dynamic hardware reconfiguration (hotplug) for hot system maintenance and changing CPU, memory, and IO bus capacity.

黒羽法男（くろばね　のりお）

Linuxソフトウェア開発統括部
所属
現在，Linuxのミッションクリティカル領域への適用性強化と顧客サポートに従事。

ま え が き

LinuxはLinuxコミュニティと呼ばれるメーリングリストにおいて，数千人～数万人規模のボランティアによる開発や使用実績作り，障害修正などの活動により，長足の進歩を続けている。とくに近年，企業のエンジニアによる開発活動も加わり，エンタプライズ分野に必要な機能が急速に開発され続けており，その多くがLinuxに取り込まれている。この結果，最新のLinuxカーネル2.6では，大規模SMP（Symmetric Multiple Processor）を効率的に動作させるためのスケジューラの機能強化やCPU間排他動作区間の大幅な削減，大規模I/O装置を取り扱うための機能強化が実施され，従来メインフレームや大型UNIXでしか構築できなかった大規模なシステムを構築するための機能が，Linuxに取りそろえられてきた。富士通は，さらにLinuxを基幹システムへ適用するために必要な機能について，Linuxコミュニティとともに開発を行い，最新のLinuxカーネル2.6の機能強化に当たってきた。

本稿では，富士通が基幹システムに必須と考え，コミュニティの一員として開発に参加している「クラッシュダンプ機能：diskdump」，「ハードウェアRAS耐性強化：MCA（Machine Check Architecture）強化」，「ハードウェア命名機能：udev」，「活性保守機能：ホットプラグ」の四つの機能について述べる。

diskdump機能

Linuxの開発は，従来ボランティアにより，機能追加や障害修正が行われてきた。この開発時にトラブルが発生した場合，ボランティアの開発者は自身で行った作業が明確であり，プログラムのどの部分がどのように動作したか大体分かるため，容易に調査することができた。また，行った作業が明確なので，再現試験を実施して障害箇所を絞り込んでいくこともできた。このため，試験を実施させながら障害を調査するツールは，非常に良く整備されている。

最近，LinuxをサーバのOSとして使用し，さらには企業の基幹システムとして利用する形態が急速に増加している。これらのシステムでは，複数の異なる業務を起動し，多数のクライアントからの要求を同時に処理するのが一般的である。このようなシステムで発生するトラブルは，トラブルが発生した時点で何を処理しているか決定できず，かつ再現性も低いものが多い。このため，トラブルが発生した時点で出力されるコンソール情報だけでは，原因不明となる場合が多くなる可能性が高い。また，サーバ運用では，システムがハングアップして何も情報を出力しないトラブルも増えている。これらのトラブル調査に有効な機能が，CPUのレジスタやメモリの内容を採取するクラッシュダンプ機能である。クラッシュダンプ機能で採取した情報からカーネルの制御テーブルの内容などを参照し，メモリの状態の矛盾を見つけ出し，障害の原因を導き出していくことが可能となる。これらの理由から，クラッシュダンプ機能をLinuxへ搭載することは，エンタプライズ向け機能として非常に重要となる。

Linuxの開発の中心となるコミュニティでは，Linuxをパーソナルユースとしている開発者が多く，ダンプ機能の有効性が，これまであまり認められてこなかった。また，異常となったカーネルの機能を利用してダンプ情報を採取するのでは，正しい情報を採取できる可能性が非常に低いのではないかと敬遠されてきた。そこで富士通では，Linuxの主要ディストリビュータの一つであるレッドハット社と開発協業し，カーネル異常（panicやoopsなど）やハングアップ時でも確実にダンプ情報を採取できるdiskdump機能を開発した。diskdumpの概念を**図-1**に，機能の特徴を以下に示す。

（1） ダンプで使用する領域を事前に獲得したり，

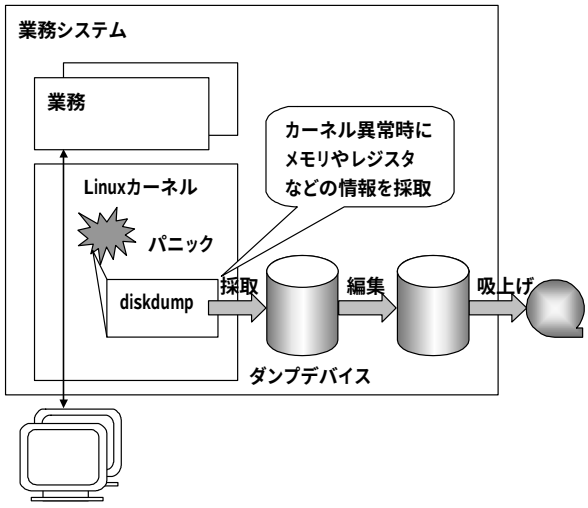

図-1　diskdumpの概念
Fig.1-Concept of diskdump.

障害発生時に他処理の動作を抑止して割込みなどの非同期事象を抑止したりするなど，トラブル発生時にカーネルの機能を極力使用しない構造としている。

(2) 出力するダンプデバイスをリセットするなどして，ハードウェアの一時的な異常状態でもダンプ採取を可能としている。

diskdump機能は，Red Hat Enterprise Linux AS（v.4 for Itanium）などのディストリビューションへの取込みを推進して一般化を図っている。また，富士通はLinuxのカーネル機能のコミュニティでの標準化に働きかけるとともに，lkdump[(1)]というコミュニティを立上げ，基幹システムのダンプ機能として広く受け入れられるように活動している。

MCA強化

基幹業務で使用されているメインフレームやUNIXサーバでは，プロセッサやメモリなどで発生したハードウェア障害がシステム全体に広がらないように，ハードウェアとOSが連携して障害を局所化し，回復する処理を行っている。

本章では，Linuxカーネル2.6で追加した，ハードウェア障害の影響を局所化し回復する機能について説明する。

本特集で紹介している基幹IAサーバ“PRIMEQUEST”は，インテル社のItanium2プロセッサを採用している。Itanium2プロセッサには従来のIAサーバに比べ，システムの可用性，信頼性を向上させるためにCPU，メモリ，チップセット，バスで発生したハードウェア障害に対する自己診断・修復機構が拡張されている。この機能により，ハードウェア故障に対し，まずハードウェア，およびファームウェアのレイヤで修復を試みる。修復が成功すれば障害によって中断されていたソフトウェアの処理は続行される。この際，OSに対してCMCI（Corrected Machine Check Interrupt）や，CPEI（Corrected Platform Error Interrupt）が通知され，OSは通知されたエラー情報をログとして記録する。ハードウェア，およびファームウェアの修復が失敗した場合は，MCAをOSに通知し回復処理を依頼する（**図**-2）。

OSは，ファームウェアが作成したエラー情報（SAL Error Record）を解析することでエラー種別ごとに回復処理を実施する。

富士通は，メインフレーム向けOS開発などで培ったノウハウを活用し，MCA強化の必要性・実装を提示しながらコミュニティと議論を続けた結果，MCA強化機能をLinuxカーネル2.6へ取り込むことに成功した。これが，メモリのECC（Error Correcting Code）マルチビットエラーに対する回復処理である。

この結果，ユーザプロセスによるメモリ読み込み時に当該エラーが発生した場合でもサーバをリブートせず，エラーの影響があったユーザプロセスのみを強制終了（sigkill）させ，かつ，エラーが発生したメモリのページを，新たな領域割り当ての対象にしないようにするという，メインフレーム並みのRAS（Reliability, Availability, Serviceability）機能がLinuxカーネルで実現できた。

なお，カーネルモードで動作中にECCマルチビットエラーが発生した場合は，カーネルで使用しているデータの保証ができないため，システムをリブートさせる。

Itanium2プロセッサで発生するMCAの中で，対処可能と判断しているそのほかのエラーとしては，PCIバス上でのパリティエラーがある。このエラーに対しては，エラーの影響を受けたI/O要求を考慮した回復処理が必要になるため，OS-MCAハンド

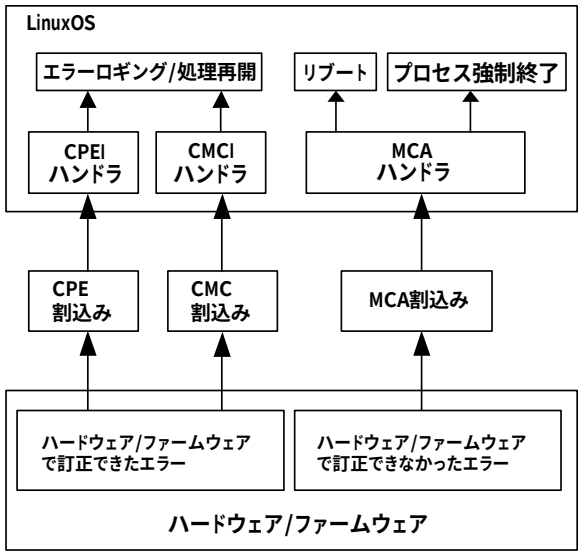

図-2　MCAの概念
Fig.2-Concept of MCA.

ラとデバイスドライバの連携が必要になってくる。現在，OS-MCAハンドラで検出したPCIバスパリティエラーをデバイスドライバに通知するためのI/Oアクセスインタフェースを検討しており，Linuxコミュニティでこの機能に関心を示しているベンダとともに標準カーネルへの取り込みに向けて活動中であり，まもなく標準カーネルに機能が取り込まれる見込みである。

udev機能

UNIX系OSでは，システムに接続されているI/O装置に対して，メジャー番号とマイナー番号と呼ばれる整数のペアを割り振り，OSはこの整数のペアによって，個々のI/O装置を識別している。これは，OSのようなプログラムにとっては管理しやすい方式であるが，システム管理者にとっては扱いにくい方式である。そのためOSがI/O装置に対して割り付けた整数のペアに対して，さらに，デバイスノードと呼ばれる特殊ファイルを対応させることにより，システム管理者はメジャー番号，マイナー番号のペアではなく，デバイスノードによってI/O装置を管理できるようになっている。LinuxもUNIX系OSの一つなので，このあたりの事情は同じであり，OSはI/O装置の管理に静的なメジャー番号，マイナー番号のペアを利用し，システム管理者はそれらに対応するデバイスノードによってI/O装置を管理している。

この方式は，システムに接続されるI/O装置の数が少ない，比較的小規模のサーバにおいてはうまく働いていたが，近年，Linuxが大規模サーバに導入されるようなり，接続されるI/O装置の数が桁外れに多くなってくると，装置の数が多すぎてメジャー番号，マイナー番号の数が不足するという問題が発生したり，装置を抜かれると，以降の装置に割り当てられる番号がずれてしまい，デバイスノードと装置の対応関係がくずれるという問題が発生したり，既存方式の延長では，新しい環境に対応していくのが困難になってきた。

最新のLinuxカーネル2.6では，メジャー番号，マイナー番号の数が不足する問題を解決するため，それぞれのフィールド幅が拡張されたので，I/O装置に対して実用上十分な数の番号を割り当てることができるようになった。また，デバイスノードとI/O装置の対応関係がくずれてしまう問題を解決するために，メジャー番号，マイナー番号のペアとデバイスノードとの関係を管理するudev機能が導入された。

udev機能は，システム管理者が定義したルールセットに従って，I/O装置に対応したデバイスノードを作成するプログラムであり，適切なルールセットを定義することによって，I/O装置に対し固定的なデバイスノードを対応付けることができる。ただし，I/O装置に対して固定のメジャー番号，マイナー番号が割り当てられるようになったわけではないので，I/O装置を固定のデバイスノードに対応付けるには，I/O装置を一意に認識できるなんらかの方法が必要となる。分かりやすい例として，LANカードにはMACアドレス（Media Access Control address）と呼ばれる48ビットの一意な識別符号が割り付けられており，これを，LANカードを一意に識別する識別子として利用することができる。同様にSCSIディスクやFC（Fibre Channel）ディスクには，VPD（Vital Product Data）と呼ばれる一意な識別符号が割り付けられており，SCSI/FCディスクを一意に識別する識別子として利用できる。

なお，信頼性やスループット向上のため，ディスクに対して複数の独立したI/O経路を設定することができ，これをマルチパス制御と呼ぶ。この場合，それぞれのI/O経路に対して独立したデバイスノードを対応付ける必要があるが，これらは同一のディスクに接続されているため，ディスクの識別子であるVPDではI/O経路を識別することができない。このようなケースでは，I/Oバス構成をI/O経路を一意に識別する識別子として使うこともできる。

例えば，PCIバスの場合，セグメント番号，バス番号，デバイス番号，ファンクション番号の四つの番号の組でバス構成を一意に識別可能であり，これらはマルチパス構成をとった場合も，I/O経路ごとに異なる番号の組が割り付けられるため，I/O経路を区別するための識別子として利用することができる。

ホットプラグ機能

高い信頼性が要求される基幹サーバでは，ハードウェアコンポーネントをモジュール化することにより，システム全体を停止することなく，モジュール

単位でハードウェアコンポーネントを交換・増設することができる。ホットプラグ機能とは，このモジュール化されたハードウェアを活性で交換・増設する機能である。また，これらのモジュールを冗長化することにより，モジュールの単一故障が発生しても，システム運用に影響を与えないようにすることができる。言い換えると，ハードウェアの自己診断機能によってモジュールが故障する予兆が検知されれば，それらが機能停止する前に，ホットプラグ機能により，予防的に交換することができる。このようなオペレーションをハードウェアの活性保守と呼ぶ。

ハードウェアをモジュール化する別のメリットとして，システムとして運用するために必要なCPU，メモリ，I/Oモジュールをグループ化し，それぞれのグループを独立したシステムとして運用することができる。このような運用をハードウェアのパーティショニングと呼ぶ。近年，サーバやストレージの仮想化技術の適用が進んでおり，TCOを削減するためにハードウェア資源を一元的にプール管理し，キャパシティーオンデマンドで割り当てる運用が現実のものとなってきたが，ハードウェアのパーティショニング技術は，このようなサーバ仮想化技術のインフラ機能として活用される。

システム運用中にハードウェア資源を増減するためには，LinuxOSに新たな機能追加が必要であり，総称してホットプラグと呼んでいる。実際には資源種別ごとに異なるホットプラグの機構が必要であり，それぞれ，CPUホットプラグ，メモリホットプラグ，I/Oホットプラグと呼ぶ。活性交換・増設可能なモジュールには，例えばCPUとメモリが一緒に搭載されているものなど，複数の異なった種類のハードウェア資源が一つのモジュールに搭載されているケースがあるため，これらの資源をモジュール単位でグループ化してホットプラグするための上位機能があり，これをノードホットプラグと呼ぶ。

現在ホットプラグ機能は，他ベンダを巻き込みながら，Linuxコミュニティで精力的に開発が進められており，その多くのコミュニティでは富士通が主要メンバとして開発に参加している。すでに，一部の機能はLinuxカーネル2.6に含まれているが，これらが本格的に利用可能になるのは，次版のカーネルになる見込みである。

むすび

Linuxは，従来からの数千人～数万人規模のボランティアを中心とした開発に加え，サーバベンダのエンジニアも加わり，機能強化が加速している。本稿では，とくに富士通がLinuxコミュニティの中で主導的な立場で開発を進め，最新のLinuxカーネル2.6に導入された強化機能について解説した。富士通は，Linuxの大規模基幹業務への適応拡大に向けて，今後も新たな機能強化に，Linuxコミュニティをリードしながら精力的に取り組んでいく。

この研究に対して「半導体アプリケーションチッププロジェクト」の一環として助成していただいた経済産業省と独立行政法人 新エネルギー・産業技術総合開発機構に感謝します。

参考文献

(1) diskdumpコミュニティのホームページ（lkdump）.
http://sourceforge.net/projects/lkdump/