



# Ottawa Linux Symposium 2006 参加報告

松原 克弥      武川 哲也  
株式会社イーゲル

2006/8/7



# Ottawa Linux Symposium (OLS) とは (1/2)

---

- x Linux技術者のためのF2Fミーティング
- x 毎年オタワで開催
  - x 今年は7月19日から22日までの4日間
- x ハッカー、熱狂的Linux崇拝者も多数
  - x 当然プレゼンテーションはLinuxで
    - x OpenOffice, MagicPoint, Acrobat, Firefox, etc.
  - x 聴衆側もWindows使いはほとんどいない
    - x Windowsがダウンすると拍手!
    - x 「本物のコンピュータ(非Windows)を持っている人はいないか？」



# Ottawa Linux Symposium (OLS) とは (2/2)

- x ソーシャルイベント?あり
  - x Welcome Reception (by Intel)
  - x Closing Reception (at BlackThorn cafe)
  - x Hacker Bike Ride
  - x Linux in the Wild Picnic
- x プレゼントあり(抽選)
- x 空いた時間で観光も可能
- x 参考:Kernel Developer Summit
  - x OLSの直前に開催
  - x 50人程度の招待者のみ出席
  - x <http://lwn.net/Articles/191137>





## Current issues

---

- x プログラムからLinuxコミュニティで今熱い(?)話題を類推
  - x Virtualization
    - x Virtualizationをサポートするための技術、機構
    - x Virtualizationの評価
    - x Virtualizationを使った応用
  - x (automatic) Test
  - x Performance measurement
  - x Tools for debugging or tuning
    - x Perfmon2, LTTng, SystemTAP, Frysk, kernel lock validator
  - x (サーバ用途の)スケーラビリティ
    - x Hugetlb, 64KB page size, CPU/memory hotplug
    - x Clustered filesystem



## *The Kernel Report (1/4)*

---

- x By Jonathan Corbet
  - x LKML(Linux Kernel ML)アーカイバLWN.netのco-funder
  - x “Linux Device Drivers 3<sup>rd</sup> Edition”の筆者の一人
- x State of the kernel (気になる部分のみ抜粋)
  - x spline()システムコールの追加 (2.6.17)
    - x ファイルコピーの効率化
    - x read/write対象のfdを直接結合することにより、メモリコピーを削減
  - x SCHED\_BATCHの追加 (2.6.16)
    - x バッチタイプのタスク向け
    - x Non-RT
    - x 常にCPU型と判断(小さいペナルティを常に与える)



## The Kernel Report (2/4)

---

- x State of the kernel (気になる部分のみ抜粋)
  - x Timeシステムコアの変更 (2.6.18)
  - x PREEMPT\_RTの一部がmainlineに取り込まれる!
    - x Light-weight robust futexes (2.6.17)
    - x Generic IRQ layer (2.6.18)
    - x 優先度継承つきfutexes (2.6.18)
  - x Lock validator (2.6.18)
    - x On the flyでロック利用ルールを解析、記憶
    - x カーネルをデッドロックする可能性があるシーケンスを警告
    - x 一貫性のない割り込み状態を警告
  - x Memory leak tracker (2.6.18)
    - x Mark-and-sweep GCライク
    - x Lost allocationのためのスタックトレース機能の提供



## *The Kernel Report (3/4)*

---

- x Virtualization
  - x Xen, UML (User Mode Linux)
  - x Linux-VServer, OpenVZ, BSD jail
  - x Issues
    - x 統一的なnamespace APIの必要性
    - x SELinuxの上 or 下のどちらで実行?
- x Security
  - x SELinux
    - x パケットラベリング、管理ツールの充実
  - x LSM
    - x 使っているのはSELinuxだけ → LSM不要論
    - x 削除 or 大幅な書き換えがあるかも → KDSでは現状維持決定?



## *The Kernel Report (4/4)*

---

- x Realtime
  - x いくつかはmainlineにとりこまれた(前述)
  - x 以下はまだだがいずれとりこまれるだろう
    - x Sleep可能なspinlock
    - x スレッドによるIRQ処理
  - x でも、使うのは”一部の”ひとだけ(という雰囲気らしい)
- x User and Kernel
  - x 多すぎるABI
    - x 膨大なシステムコール
    - x /proc
    - x /sys
    - x netlink



## *Embedded Linux BOF*

---

- x By Tim R Bird
  - x CELF AG Chair
  - x SONY
- x 組み込み分野に影響がありそうなカーネルupdateを紹介
- x CELF活動の説明
  - x Bloatwatch
    - x バージョン別のカーネルサイズの変化を可視化
  - x Config weight size test
    - x CONFIG毎のメモリ使用量を自動計測
  - x CABI (CPU資源管理)
    - x RTタスクに対するCPUリソース制限
    - x Non-RTタスクに対する優先度ブースト



# AppArmor Application Security BOF

---

- x By Crispin Cowan
  - x Novell
  - x StackGuard, LSMの発案者
  - x 元 Immunix CTO(ImmunixはAppArmorとともにNovellが買収)
- x システム全体を統括すると(SELinux)いうよりも、各アプリケーションを(ネットワーク経由の)外敵から守るというスタンス
- x パス名でサブジェクト、オブジェクトを識別
  - x 各オブジェクトにラベルづけするよりも簡単
  - x (私的感想) busyboxは大丈夫?
- x ポリシーファイルの自動生成が可能
  - x 実際に2分程度でApacheのポリシーファイルを作ってしまった。



# *The Need for Asynchronous Network I/O (1/2)*

---

- x By Ulrich Drepper
  - x RedHat
- x 背景
  - x ネットワークH/Wの進化に伴い、バイト転送を基本とするSocket APIは非効率なものとなった。
  - x POSIX\_AIOはネットワークI/Oではない
  - x O\_ASYNC for sockets ほしいが、非同期I/Fのための効率的なイベントハンドリングが必要
  - x ユーザバッファへの(からの)直接的なI/O



# The Need for Asynchronous Network I/O (2/2)

---

- x 既存アプローチ
  - x メモリロック: read()時に指定したバッファメモリをロックして、NICから直にコピー
    - x 特権オペレーション、高価、ページサイズ単位で非効率
  - x epoll\_wait()
    - x ファイルディスクリプタに対してのみ使用可
    - x 同期プリミティブやメッセージキュー、AIO要求、シグナルには使えない
  - x SOCK\_SEQPACKETプロトコルのPF\_EVENT
    - x イベントキューに上限あり
- x 提案
  - x ユーザレベルでイベントを扱う機構「イベントチャネル」の新設
  - x ユーザレベルでNetwork I/Oに対する直接的DMA指示API



## *Hrtimers and Beyond: Transforming the Time Subsystems (1/2)*

---

- x By Thomas Gleixner
  - x 組み込み系
- x タイマ機構の遍歴
  - x Double linked .. -> UTIME -> Timer Wheel -> HRT -> hrtimers
- x Hrtimers
  - x タイマは、“red-black tree”により管理
    - x  $O(\log(N))$
    - x Expiring time順にソート
    - x CPU毎のリスト
  - x 各々のベースクロックに応じてキューを分割
  - x ベースコードはtick driven
  - x 時間はktime\_tというナノ秒ベースの構造体で保持

x



## *Hrtimers and Beyond: Transforming the Time Subsystems (2/2)*

---

- x 時間の表現: `ktime_t`
  - x 32ビットと64ビットマシンに最適化
  - x 64ビットCPUではplain, 32ビットでは、秒とナノ秒のペアで処理
  - x 64ビットベースの加算、減算、比較が可
- x Hrtimersのユーザ
  - x `Nanosleep()`, `itimer()`, POSIX timers, `timed futex`オペレーション
- x Dynamic Tick
  - x idle中のtickを抑制することで消費電力やコンテキストスイッチ・オーバヘッドに貢献
  - x Issue: 次のexpiring timeをどうやって見つけるか
    - x 従来: 最悪、膨大なハッシュをすべて検索
    - x Hrtimers: red-black treeの最初のエントリが次のexpiring time



# Improving Linux Startup Time Using Software Resume

---

- x By Hiroki Kaminaga
  - x SONYの技術者
  - x CELFから投稿した日本人で唯一acceptされた
- x カーネルやアプリケーションのメモリエイメージが変わらない組み込み系に最適
- x SWSUSP2を使って、ブート後のサスペンド・メモリエイメージを作成し、不揮発メモリに記録
- x 改造したブートローダにより、サスペンドイメージを直接実行メモリ位置にロードし、実行を再開
  - x 通常のソフトウェアサスペンドは、カーネルブートの最終段でロード
- x 電源投入からアプリ実行開始まで11秒→5秒に削減
- x ARMのみ実装



# Extending RCU for Realtime and Embedded Workloads (1/2)

---

- x By Paul E McKenney
  - x IBM
- x (PREEMPT\_RT適用環境のための)リアルタイム性を向上させたRCU”Realtime RCU”の提案
  - x よりオーバヘッドの少ないrcu\_read\_lock(), rcu\_read\_unlock()
    - x Read-side危険領域ではpreemption禁止(atomic操作、メモリバリア、割り込み禁止)により大きな遅延発生 → atomic操作とメモリバリアを削除
  - x よりスケラブルなgrace-priod検出
    - x grace-priod検出コードで一つのグローバルキューを使用←キューへのロックがボトルネックに
  - x コールバック遅延とスループットとのバランス
  - x 構造体毎メモリアーバヘッドの削減
    - x call\_rcu()に渡される構造体へのポインタ8バイトを削減



## *Extending RCU for Realtime and Embedded Workloads (2/2)*

---

- x
- x Read-side危険領域での優先度ブースト
  - x 「-rt」環境では、割り込み処理をpreemptできるので、RTタスクがCPUの占有が可能
  - x 優先度をあげて、危険領域処理が無限に遅延されるのを防ぐ
  - x 完全に実現できていない ← (CABIのような)CPU資源管理が有用



# Measuring Resource Demand in Linux (1/3)

---

- x By Rik van Riel
- x Linuxにおける資源使用の様子を見るテクニックを紹介
- x 背景
  - x 要求された資源 ≠ 使った資源
  - x 経験を積んだ管理者はボトルネックを見つけることができる。
- x CPU時間
  - x User & system, idle (Linux創世記から存在)
  - x IOwait, irq, softirq (2.6から登場)
  - x Steal (仮想マシンのみ)



# Measuring Resource Demand in Linux (2/3)

---

- x CPU steal time
  - x タスクを実行している仮想CPUが使っている時間
  - x 仮想CPU自身が走っている時間ではない
  - x Hypervisorによりpreemptされる
  - x
- x CPU時間の解析
  - x Idleとiowait時間が大きい
    - x CPU性能は十分
  - x steal時間が大きい(idle, iowaitは小さい)
    - x VMが多すぎる
  - x steal, idle, iowaitとも小さい
    - x 負荷に対してCPU性能が低い



# Measuring Resource Demand in Linux (3/3)

---

- x メモリ性能
  - x 最近はき出された(evicted)メモリの再利用率を計測
  - x Evictedメモリの再フォルト距離(refault distance)を計測
- x Refault distance
  - x あるページがフォルトしたら、そのページが最近evictedされていたかをチェック
  - x 上記ページ数がrefault distance
- x Refaultの回避
  - x メモリの増設 → iowait時間の軽減、システム性能向上



# *Linux as Hypervisor – An Update*

---

- x By Jeff Dike
  - x Intel
- x Linuxをhypervisorとして利用する観点からLinux機能の遍歴、今後の改善ポイントを紹介
  - x VMはUMLを想定
- x 背景
  - x VMは新種のworkload
  - x 命令のエミュレートはより多くのCPU性能を必要とし、I/Oスピードも要求するかもしれない。



# Linux as Hypervisor – An Update

(1/3)

---

## x 過去

### x Ptrace

- x システムコールの仮想化に必要
- x nullfyシステムコール:i386の初期システムではパッチが必要だった
  - x 割り込む前にシステムコール番号が保存されてしまっていた
- x PTRACE\_SYSEMUの追加
  - x システムコール発行前のみインターセプト
  - x コンテキストスイッチを半分に

### x AIO and O\_DIRECT

- x 非同期I/Oがないと、UML上の1タスクがI/O処理を始めた瞬間全体がブロック
- x 2.6でAIOが登場するまでは、I/Oスレッドを作って対応
- x O\_DIRECTの登場により、host OsとUMLの両方でキャッシュすることを回避
- x ただし、現在はO\_DIRECTでのみAIO可能



# Linux as Hypervisor – An Update

(2/3)

---

- x 現在
  - x カーネル仮想化インフラ
    - x VServerのような軽量コンテナのための対応
    - x UMLのシステムコールをinterceptionなしで実行可能に
  - x FUSE
    - x UML filesystemをホスト上でmount可能に
    - x UML上のread(),write()コールがホスト上で直接実行可能に
  - x remap\_file\_pages()
    - x 非線形ファイルマッピングを実現
    - x VMA操作が性能に影響
    - x 1ページ1VMA(vm area struct) → 複数ページ1VMAに再構成



# Linux as Hypervisor – An Update

(3/3)

---

## x 未来

### x AIOの拡張

- x Bufferedデータに対するAIO

### x AIO mmap

- x Mmapはメモリ節約になるがTLBフラッシュのオーバーヘッドを伴う
- x TLBフラッシュ・オーバーヘッドの隠蔽に有効

### x メタデータに対するAIO

- x 現状、stat()のようなメタデータアクセスはブロックする

### x リモートアドレス空間への操作

- x 同じプロセス内にUMLカーネルとUMLタスク
- x /proc/mm
  - x Linusはきれい
- x mm\_indirect
  - x Linusが提案



# *Reliable and Portable Multimedia File System (1/3)*

---

- x 概要
  - x データベースを使ったファイルシステムについての説明
  - x eXtensible Portable Reliable Embedded Storage System (XPRESS)
  
- x DBとしてBerkeley DBを使用し、inodes、directories、空きスペースなどのメタデータをDBで管理



## *Reliable and Portable Multimedia File System (2/3)*

---

- x ファイルシステムにDBを使う利点
  - x XPRESSのフロントエンドをユーザ空間に構築
    - x 他のシステムに移植が容易 (Portable)
    - x メンテナンスが容易 (Maintainability)
  - x ジャーナルの代わりに、既存のトランザクション処理が利用できる (Reliable)
  - x ファイルシステムのバージョンアップが容易
    - x DBのスキーマを変更するだけ



# *Reliable and Portable Multimedia File System (3/3)*

---

- x 高速なファイル編集 (Multimedia)
  - x ビデオなどの編集では、ストリームの一部分を削除する必要があるが、このような処理を高速で行うために、XPRESSでは論理アドレス空間と物理アドレス空間を扱う
    - x ストリームの一部を削除すると、論理アドレス空間が編集されるだけで、物理的にデータの一部を削除して前に詰めるなどの処理は発生しない
    - x 論理・物理アドレス空間のマッピングを管理している



# Fully Automated Testing of the Linux Kernel (1/3)

---

- x 概要
  - x カーネルのテストの完全な自動化に関する解説
  
- x なぜ自動化する必要があるのか？
  - x 退屈だから
  - x 2.6では変更点が多い
  - x 非常に広い範囲にLinuxは使われており、昔ながらのバグ報告では対応できない



## *Fully Automated Testing of the Linux Kernel (2/3)*

---

- x 自動化されたテストは、リリース前のカーネルに対して行われる
  - x 一般のユーザがバグの被害者になることを避ける
  - x なるべく早い段階でテストすることで、依存するバグを生成しないようにする
- x 現状、1日2回、自動テストがカーネルに対して行われている



## *Fully Automated Testing of the Linux Kernel (3/3)*

---

- x テストはPythonで書かれている
  - x 例外処理が扱える
  - x 管理が容易
  - x モジュールが豊富
  
- x 現状より、さらに上流でのテストをする必要がある
  - x 企業もテストへ参加することが望ましい
  - x バグで苦しむ前に、バグを減らすことに協力する方が建設的



# Xen 3.1 and the Art of Virtualization (1/4)

---

- x 概要
  - x Xenの現在の開発状況の報告
  
- x 3.xでの代表的な新機能
  - x Hardware Virtual Machine (HVM)
    - x VT-x、SVMを仮想化するためのレイヤで、両者を同様に扱うことが出来るようになる



- x MiniXen
  - x Linuxを改さずに直接ハードウェアに命令を伝える仕組み。
  - x ネイティブパフォーマンスに近い値が出る。
  - x Xen用に編集された1つのカーネルを起動可能
  
- x IA64対応状況の進歩
  - x 昨年より大幅に進捗があった
  - x x86レベルに追いつきつつある



# Xen 3.1 and the Art of Virtualization (3/4)

---

- x PowerPCへの対応状況
  - x ポーティング作業を進めているところ
    - x PowerPC 970が対象
  - x PowerPC 970でも、hypervisorが使えるもの (IBM pServer) と、使えないもの (Apple G5) がある
    - x hypervisorの有無に関わらず動作するようにしようとしている



- x □ロードマップ
  - x NUMAのサポート
  - x IOMMUのサポート
  - x ユーザ空間から直接アクセスを許可するようなデバイス(主にネットワークデバイス)のサポート
  - x Virtual Framebufferのサポート
    - x 仮想マシンに対してVNCで接続できるようになる



# *Linux Bootup Time Reduction for Digital Camera (1/6)*

---

- x 概要

- x スチールカメラが使用可能になるまでの時間を以下に短くするかを解説



# Linux Bootup Time Reduction for Digital Camera (2/6)

---

- x スチールカメラが使用可能になるまでの概要
  - x Boot Loader
    - x システムの初期化
    - x RAMにカーネルイメージをコピー
  - x Kernel Initialization
    - x カーネルの初期化
    - x デバイスドライバの初期化
  - x Application Initialization
    - x RCスクリプトの実行
    - x アプリケーションの実行
    - x プレビューモード(スチールカメラが使用可能)



# Linux Bootup Time Reduction for Digital Camera (3/6)

---

- x Boot Loader
  - x NOR型フラッシュメモリなら、XIPを使う
    - x XIP
      - x 実行ファイルのText部分を、実行時にRAMにコピーしない
  - x Root File System (RFS)をRAMにコピーする際には、コピーと解凍に時間がかかるので、cramfsを使う(カーネルイメージのみ)
    - x cramfsを使うと、必要なファイルだけをコピー・解凍するため起動時間が短くなる。ただし、読み込み専用でしか使えない



# *Linux Bootup Time Reduction for Digital Camera (4/6)*

---

- x Kernel Initialization
  - x Init Scriptの見直し
    - x /etc/inittabなど
  
- x Application Initialization
  - x 必要なときにロードの方針で作成
  - x fork()の必要のあるタスクも、必要になったら生成
    - x 初期化時に呼ぶと、起動時間が長くなる



# Linux Bootup Time Reduction for Digital Camera (5/6)

---

## x その他

- x loops\_per\_jiffiesの値をプリセットしておく
  - x [http://tree.celinuxforum.org/CelfPubWiki/\\_e8\\_b5\\_b7\\_e5\\_8b\\_95\\_e6\\_99\\_82\\_e9\\_96](http://tree.celinuxforum.org/CelfPubWiki/_e8_b5_b7_e5_8b_95_e6_99_82_e9_96)
- x 起動時にコンソールにログが出力されないようにする
  - x ログを出力する時間をもたない
- x カーネルモジュールは動的に後からロードする
  - x USB、MPEG、STROBE、WDT、TVなど



# *Linux Bootup Time Reduction for Digital Camera (6/6)*

---

- x kmallocやmallocは、ブート後に行うか、初めからメモリを割り当てておく
- x サスペンド・リジュームを使う



# *(Tutorial) Porting a 2.4.20 Character Driver to 2.6.15 (1/4)*

---

- x 概要

- x Mark Grossの経験を元に2.4系のデバドラを2.6系にポーティングする方法についての解説

- x

- x 基本は、コンパイルしてエラーを取っていく

- x

- x 細かい注意は以下のURLを参照

- x <http://www.intel.com/cd/ids/developer/asmo-na/eng/301689.htm>



## *(Tutorial) Porting a 2.4.20 Character Driver to 2.6.15 (2/4)*

---

- x ioctlはsysfsに置き換える
  - x udevに対応するためにもsysfsは使う
  - x (置き換えの実装例も前述のURLを参照)



# (Tutorial) Porting a 2.4.20 Character Driver to 2.6.15 (3/4) ~参考~

---

- x ユーザスペースとのデータのやり取りについて
  - x 様々な手法が存在している
    - x ioctl
    - x sysfs
    - x proc
    - x relayfs
    - x debugfs
    - x foofs
    - x netlink
    - x setsockopt()

# igel (Tutorial) Porting a 2.4.20 Character Driver to 2.6.15 (4/4) ~参考~

---

- x Linusは、sysfsはioctlに比べて複雑なトランザクション処理で使い難いが、多くの点で優れていると考えている
- x Kernel Summit 2006では、ioctlについて話し合いがもたれた
  - x 適切なインターフェースについてガイドラインが必要との結論
  - x 誰が書くのかが今後の問題



# *Native POSIX Threads Library (NPTL) Support for uClibc (1/4)*

---

## x 概要

- x glibcより軽量でEmbedded向きのuClibcでNPTLを実装したという報告
  - x (現在のところMIPSのみ)
- 
- x uClibcは既にLinuxThreadを実装済み
    - x NPTLはLinuxThreadより重い(ファイルサイズ、メモリ使用量)が、POSIX準拠で効率も良い
    - x Embeddedのリソースが向上してきているので、NPTLを実装する意味が出てきた



# *Native POSIX Threads Library (NPTL) Support for uClibc (2/4)*

---

- x uClibc上のNPTLの実装
  - x
  - x Thread Local Storage(TLS)は、4種類(error, h\_error, \_res, RPC\_VARS)だけをサポート
  - x 非同期I/Oはサポートしていない



# *Native POSIX Threads Library (NPTL) Support for uClibc (3/4)*

Static Library	glibc [bytes]	uClibc [bytes]
libc.a	3 426 208	1 713 134
libcrypt.a	29 154	15 630
libdl.a	10 670	36 020
libm.a	956 272	248 598
libnsl.a	161 558	1 100
libpthread.a	281 502	250 852
libpthread_nonshared.a	1 404	1 288
libresolv.a	111 340	1 108
librt.a	79 368	29 406
libutil.a	11 464	9 188
<b>TOTAL</b>	<b>5 068 940</b>	<b>2 306 324</b>

Static Library Sizes (glibc vs. uClibc)



# *Native POSIX Threads Library (NPTL) Support for uClibc (4/4)*

---

Shared Object	glibc [bytes]	uClibc [bytes]
libc.so	1 673 805	717 176
lib.so	148 652	35 856
libcrypt.so	28 748	13 676
libdl.so	16 303	13 716
libm.so	563 876	80 040
libnsl.so	108 321	5 032
libpthread.so	120 825	97 189
libresolv.so	88 470	5 036
librt.so	45 042	14 468
libutil.so	13 432	9 320
<b>TOTAL</b>	<b>2 807 474</b>	<b>991 509</b>

Shared Object Sizes (glibc vs. uClibc)



# Automatic System for Linux Kernel Performance Testing (1/3)

---

## x 概要

- x クライアントサーバ型のカーネルテストシステムについての報告
- x 現在は、x86, alpha, ppcについて開発中
- x ユーザの様々な環境下でのテスト結果を集計し、パフォーマンスの測定や問題特定に役立てる
- x
- x パフォーマンスは、特定のテスト環境下だけでは測定し難い要素
  - x さまざまな環境下で測定されることで、正しい結果を知ることが出来る



# Automatic System for Linux Kernel Performance Testing (2/3)

---

## x システム構成

- x ユーザはテストを行いたいカーネルを指定すると、自動的にダウンロード、コンパイル、インストール、パフォーマンステストを行い、結果をサーバに送信
- x 開発者は、パフォーマンステストの結果をWebインターフェースより解析



# Automatic System for Linux Kernel Performance Testing (3/3)

---

- x いくつかの未解決の問題
  - x バグの含まれたカーネルをブートすることで、ユーザの環境に問題を起こす可能性がある(ファイルシステムが破壊されるなど)
  - x テスト結果に信頼性が置けない可能性がある(悪意のユーザの存在)



# *(Tutorial) Reverse Engineering USB Driver (1/6)*

---

## x 概要

- x 他のOS用のデバドラの存在するUSB機器について、Linux用のデバドラを開発する方法を説明
- x リバースエンジニアリングは法的に問題がある
  - x このチュートリアルは「教育目的」



# (Tutorial) Reverse Engineering USB Driver (2/6)

---

- x USB機器は便利だが、Linux用のデバイスドライバが公式に配布されていないことが多い
  - x ベンダーの言い訳
    - x マーケットが小さい
    - x 誰もLinuxなんて使っていない
- x スペックも公開されていない
  - x 公開されたとしてもライセンスの問題
    - x NDAを結べ、etc



# *(Tutorial) Reverse Engineering USB Driver (3/6)*

---

- x ベンダーやコミュニティのサポートを待つしかないのか
  - x 自分でどうにかできる



# *(Tutorial) Reverse Engineering USB Driver (4/6)*

---

- x 開発のステップ概要
  - x 他のOSで動作するデバドラを入手する
  - x そのデバドラが何をしているかをキャプチャする
    - x USBのスペックを読む(<http://www.usb.org/developers/docs/>)
    - x VMware
    - x USB Snoopy (Windowsで動作するキャプチャソフト)
    - x usbmon (Linuxで動作、USBの通信をキャプチャ)
  - x 同じことをLinux上で行う
    - x libusb (ユーザ空間でデバドラを書くためのライブラリ)



# *(Tutorial) Reverse Engineering USB Driver (5/6)*

---

## x いくつかの問題

- x キャプチャした際に何が起こっていたのか、ログとの関連性を把握しなければならない
- x 複数のメッセージに分割されてやり取りされているデータが存在する
- x 必要なデータを得るためには、非常に長いテキストファイルのログをパースし、場合によっては、結合しなければならない



# *(Tutorial) Reverse Engineering USB Driver (6/6)*

---

- × EthrealライクはGUI
  - × このチュートリアル of 発表者であるEric Prestonが作成したEthrealライクなGUI
  - × 以下で公開される予定(?)
    - × <http://www.linuxmontreal.com/projects/usb/reveng/>



## Keynote Address Keynote

---

- x By Greg Kroah-Hartman
  - x Novell
  - x “Linux Device Drivers 3<sup>rd</sup> Edition”の著者の一人
  - x Linux Journalのエディタ
- x [Mythsに関する嘘と真実をあぶりだしながらLinuxのすばらしさを解説を巡る神話とウソとホント)
- x
  - x Linuxは対応しているデバイスが少ない?
    - x USB2とbluetoothに対応したのはLinuxが最初
  - x ロードマップがない?
    - x “進化”にロードマップは必要ない。
  - x APIが安定しないからベンダーがドライバを書かない?
    - x documentation/stable\_api\_nonsense.txtを見よ。



## Keynote Address Keynote

---

- x 後半は、closed-sourceな(バイナリモジュールのみが提供される)カーネルモジュールの話題に
  - x ソースを公開しないモジュールはillegal
  - x Novelの公式発表として、今後closed-sourceのモジュールは配布しない(RedHatも同じようなスタンス)
  - x 質問で、ユーザレベルのドライバはどう思うかという質問あり
    - x “Great””I love it.”
    - x StaticなAPIを取り決めて、ユーザランドでデバドラを実装することは何ら問題ない。



- x OLS Proceedings
  - x <http://www.linuxsymposium.org/2006/proceedings.php>
- x “The Kernel Report” by Jonathan Corbet のスライド
  - x <http://lwn.net/talks/ols2006/>
- x オタワLinuxシンポジウム レポート by David-Graham
  - x <http://opentechpress.jp/developer/06/07/24/0223208.shtml>
  - x <http://opentechpress.jp/developer/06/07/25/0251249.shtml>
  - x <http://opentechpress.jp/developer/06/07/27/0219220.shtml>
  - x <http://opentechpress.jp/developer/06/07/28/0243215.shtml>