



# OLS 2006 report: Dump discussion

2006 / 9 / 11

NTT Data Intellilink Corporation  
Fernando Luis Vázquez Cao



# Who am I ?

- LKDTT (Linux Kernel Dump Test Tool) maintainer
- MKDump (Mini Kernel Dump) co-maintainer
- LKDTT (Linux Kernel Dump Test Tool) のメンテナー
- Mkdump (Mini Kernel Dump) の開発者



# Agenda

1. Ottawa Linux Symposium (OLS):  
Expectations
2. Pres. : Evaluating Linux Kernel  
Crash Dumping Mechanisms
3. BOF : Reinitialization of Devices  
after a Kexec Reboot
4. OLS: Conclusion



# ***1. OLS : Expectations***



## 1.1. What did I want from OLS?

- Increase awareness of Kdump reliability issues as well as deficiencies in the kernel crash detection mechanisms

⇒ **Presentation**: *Evaluating Linux Kernel Crash Dumping Mechanisms*

- Kdumpの信頼性の問題およびカーネルの障害検出機能の不足について意識を喚起する

⇒ **発表**: 「Linuxカーネルにおけるクラッシュダンプの仕組みの評価」



## 1.1. What did I want from OLS? (cont / 続き)

- Discuss the device initialization issues after a kexec boot, which were not being addressed at the moment due to lack of consensus

⇒ **BOF**: *Reinitialization of devices after a kexec reboot*

- 合意の欠如の為に疎かになっていた、Kexecリブート後のデバイスの再初期化の問題について議論する

⇒ **BOF**: Kexecリブート後のデバイスの再初期化



## ***2. Pres. : Evaluating Linux Kernel Crash Dumping Mechanisms***



## 2.1. Kernel crash dump

- Snapshot of the system at the time that the kernel crashed
  - Includes: memory image, register contents
  - Essential for post-crash analysis of the kernel
  
- カーネルが故障した瞬間のスナップショット
  - 内容: メモリの情報、レジスタのステータス等
  - クラッシュの原因を究明する為の最も重要な材料



## 2.2. The need for testing

- Kernel crash dumping solutions available for Linux: LKCD, diskdump, Kdump (in mainline kernel)
- Having a crash dumping mechanism does not guarantee that we can obtain a dump under any crash scenario
  - Need estimate of the probability of capturing a dump
- Linuxにおけるカーネルクラッシュダンプ取得の仕組み: LKCD、diskdump、kdump(本家カーネル)
- カーネルダンプ取得機能があるからといって、確実にダンプが取れるわけではない
  - ダンプ取得の確実さを測定し、検証すべき



## 2.2. The need for testing (cont / 続き)

- Fair comparison of existing crash dumping solutions
  - Standard testing procedures have to be established
  
- 全てのダンプツールを公平に比較しなければならない
  - 標準化されたテストツールが必要



## 2.3. Traditional crash dump testing

- A test kernel module artificially causes the kernel to crash
  - By directly invoking “panic” or “BUG”, dereferencing a null pointer, etc.
  
- テスト用のカーネルモジュールを利用し、そのモジュールの延長で panic や BUG 等と呼ぶことによってカーネルのクラッシュを引き起こす



## 2.3. Traditional crash dump testing (cont / 続き)

- **Problem:** the coverage of the tests is very limited
  - All the crash dumping solutions seem to be very close in terms of reliability
  - ⇒ This seems to contradict theory
- **Reason:** the state of the HW and the execution context is not being taken into account
- **問題:** 網羅性がない
  - 全てのクラッシュダンプ取得の仕組みが同等の高信頼性を持つように見えてしまう
  - ⇒ このテスト方法は怪しい
- **原因:** HW状態と実行コンテキストを考慮せず



## 2.4. LKDTT (Linux Kernel Dump Test Tool)

- LKDTT is a tool that forces the system to crash by artificially creating crash scenarios
  - The necessary hardware and load conditions can be recreated
  - The execution context can be precisely defined
  - ⇒ It is possible to perform realistic tests
  
- LKDTTはカーネル中の障害を人為的に発生させる仕組み
  - 特定のHWの状態が再現できる
  - クラッシュ時の実行コンテキストが設定できる
  - ⇒ 現実的なテストが可能



## 2.4. LKDTT (cont / 続き)

### ■ LKDTT test results

- Kdump proved to be much more reliable than traditional in-kernel crash dump solutions
- But several deficiencies in Kdump were revealed too

### ■ LKDTTによるテストの結果

- Kdumpが従来のダンプ取得の仕組み(LKCD、diskdump等)より優れている
- しかし、Kdumpにはいくつかの欠点が発見された



## 2.5. Kdump issues

### ■ Crash detection

- Certain crashes with interrupts enabled are not detectable
- There are no reliable stack overflow detection mechanisms

### ■ 障害検出

- 割り込み可能な状態でのハングが検出できないことがある
- 既存のスタックオーバーフロー検出の仕組みは信頼できない



## 2.5. Kdump issues (cont / 続き)

### ■ Reliability

- Kdump is very vulnerable to stack overflows
- Sometimes the dump capture kernel crashes

### ■ 信頼性

- スタックオーバーフローが発生した場合は kdump が失敗し易い
- クラッシュダンプ取得用のカーネルがクラッシュすることがある



## 2.6. Impact

- LKDTT revealed serious reliability issues in Kdump and deficiencies in the kernel's crash detection mechanisms
  - The necessity of the fixes and enhancements proposed during the presentation was recognized
    - ✗ Some of them have already found their way into the mainline kernel and there is more to come
  
- LKDTTによって、Kdumpの信頼性の問題やカーネルの障害検出機能の欠如が発見された
  - 発表で提案した修正や機能拡張は認められた
    - ✗ 一部は本家のカーネル又はテストカーネルに採用
    - ✗ 残りのパッチは整理とテストができ次第投稿する



## 2.6. Impact (cont / 続き)

- In addition to current users such as IBM, HP and Intel have shown their interest in LKDTT
  - More LKDTT users means more testing which, in turn, should lead to a more reliable Kdump
  - It seems that IBM has even started the test automation effort
  
- 現ユーザ (IBM等)に加えて、HPとIntelにもLKDTTを利用してもらえることになった
  - LKDTTによるテストの数が増えれば増えるほど、Kdumpの信頼性に反映される
  - IBMはLKDTTのテストを自動化したそう



## ***3. BOF : Reinitialization of devices after a kexec reboot***



## 3.1. Device reinitialization

- In the event of a crash the state of the devices is unknown
  - The dump capture kernel needs to initialize the underlying devices again  $\Rightarrow$  device reinitialization
  
- 異常が発生した際にデバイスの状態が不明
  - kdump カーネルが新しくデバイスを初期化しなければならない  $\Rightarrow$  デバイスの再初期化



## 3.1. Device reinitialization (cont / 続き)

- When the dump capture kernel boots devices might be operational or in an unstable state
  - Device drivers might not be able to handle those cases
    - ✗ The kernel fails to initialize a device and crashes

⇒ **The crash dump capture kernel crashes**

- Kdump カーネル起動時に、デバイスが暴走している可能性が高い

- デバイスが予期しない状態になっている
  - ✗ kdump カーネルがデバイスの初期化に失敗する恐れがある

⇒ **クラッシュダンプ取得用のカーネルがクラッシュする**



## 3.2. Device reinitialization: Solutions

4 possible solutions were proposed and discussed:

1. **Black list of devices**
2. **Device reset** (device soft-reset, PCI bus reset)
3. **Device configuration restoration** (save the configuration performed by the BIOS and pass it to Kdump)
4. **Driver hardening**

4つの解決方法を提案した:

1. **デバイスのブラックリスト**
2. **デバイスリセット**
3. **デバイスの設定のリストア**
4. **デバイスドライバの改善**



## 3.3. BOF : Conclusion

- Start by creating a **black list of devices** that are known to have problems
  - **Maintainers' reaction**: I do not want to see my driver in such a (*censored*) list! I will fix any bug that might be found!
  
- きちんと再初期化できない**デバイスのブラックリスト**を作ることから始める
  - **メンテナーの反響**: 私が管理しているデバイスドライバがブラックリストに入ってほしくないなので、何かあったら直しますぞ



## 3.3. BOF : Conclusion (cont / 続き)

- Test kexec and kdump using **LKDTT** and fix problems as they are found by **hardening the drivers**
  - It is necessary to add new test cases to LKDTT covering the device reinitialization issues
  
- **LKDTT**で kexec と kdump をテストしながら**ドライバを改善**していく
  - デバイスの再初期化に関するテストケースを追加する必要がある



## 3.3. BOF : Conclusion (cont / 続き)

- Sometimes hardening the driver does not suffice
  - Consider alternatives such as **soft-resetting** the device or **restoring its original configuration**
    - ✗ These two approaches are considered overkill though, so they should only be used as the last resort
  
- デバイスドライバを修正しても再初期化の問題が解決ができない場合がある
  - **デバイスリセット** 又は **デバイスの設定のリストア** の手段を検討
    - ✗ 上記の二つの手段は大袈裟と思われる為、最後の手段と考えるべき



## ***4. OLS: Conclusion***



## 4.1. BOF : Conclusion

- Sometimes to get things started or get the ball rolling faster F2F discussion is indispensable  
Example: During the kexec BOF, the kexec maintainer, the kdump maintainer, the ACPI maintainer, embedded systems developers, and device drivers maintainers gathered and agreed on a workable solution
- 取り組みを始めたたり開発や決定を円滑したりする為、F2F で議論するのは非常に大事である  
例: 今回の BOF では kexecのメンテナー、kdumpのメンテナー、ACPIのメンテナー、組込システムの開発者、デバイスドライバのメンテナー等が集まって、実現可能な解決策について進め方を決めた



## 4.2. Presentation : Conclusion

- To increase the awareness of a certain issue presenting it in a public forum such as OLS is a great help

Example: After the OLS presentation the number of companies using LKDTT to test Kdump increased

- Linuxカーネルに関する問題について注意を喚起したい場合は、OLS は最も適切な所である

例: OLS の発表をきっかけとして、LKDTT のユーザを増やせることになった



## 4.2. OLS: Opportunities

- Present your work to an international audience of Linux experts ⇒ **Presentations**
- Discuss kernel issues that have stagnated or are not being addressed properly (if at all) ⇒ **BOF**
- 世界のLinuxエキスパート達を相手に自分のプロジェクトを紹介する ⇒ **発表**
- 沈滞した、又は取り組まれていない問題について議論する ⇒ **BOF**



## 4.2. OLS: Opportunities / 機会 (cont / 続き)

- Offer an in-depth practical introduction to your work for prospective users/contributors ⇒ **Tutorials**
- Discuss specific issues and exchange ideas with your peers without risking your face ⇒ **Corridor conversations**
- ユーザと貢献者を増やす為、実践徹底入門を提供する ⇒ **チュートリアル**
- 特定の問題について議論したり意見を交換したりする ⇒ **捕まえ立ち話**



Thank you for your  
attention

ご清聴有難う御座いました

連絡先: [fernando@intellilink.co.jp](mailto:fernando@intellilink.co.jp)

[fernando@oss.ntt.co.in](mailto:fernando@oss.ntt.co.in)



**No excuse not to  
come to next  
year's OLS !**