RAS Round-up From the IBM Linux Technology Centre

LinuxTag 2002

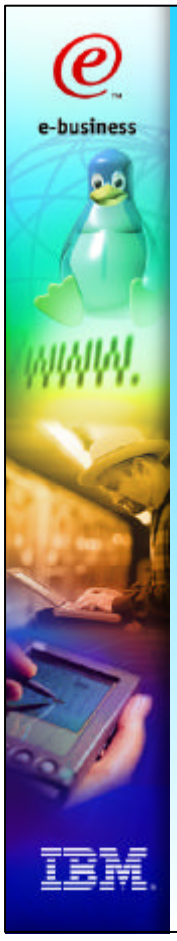
Karlsruhe

Richard J. Moore C.Eng, MIEE, MIEEE, BSc

richardj_moore@uk.ibm.com

IBM Linux Technology Centre

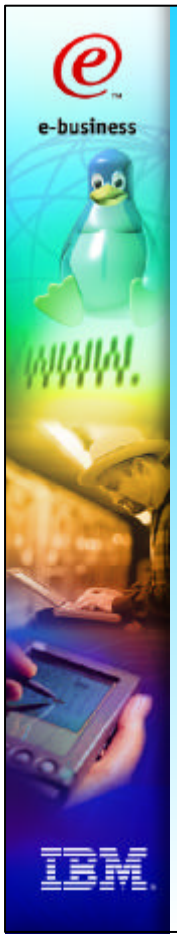
10th May 2002 (v6) lt-330588670



Topics

1. Dynamic Probes
2. Kernel Hooks (GKHI)
3. Linux Event Logging for the Enterprise
4. Flexible Dump
5. System Trace
6. Community Adoption
7. Miscellaneous
8. What's Next
9. The Team - Contacts

Dynamic Probes (DProbes)



1.1 Dynamic Probes - What is it?

- **Low-level Universal Debugger**
  - ➡ **Operates in extreme conditions**
  - ➡ **Kernel/User, Interrupt/Task, Code/Data**
  - ➡ **For Service/Support Engineers on Production Systems**
  - ➡ **Monitors Low-level System Resources**
  - ➡ **Dynamic & Fully Automated**
  - ➡ **Trigger/Enabler for:**

    KDB,

    LKCD,

    LTT,

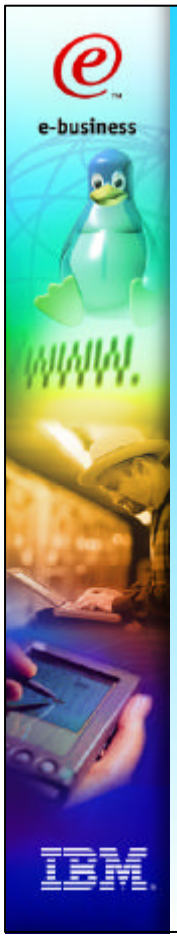
    evlog,

    Core Dump,

    Syslog, etc

1.2 Dynamic Probes - Where Used?

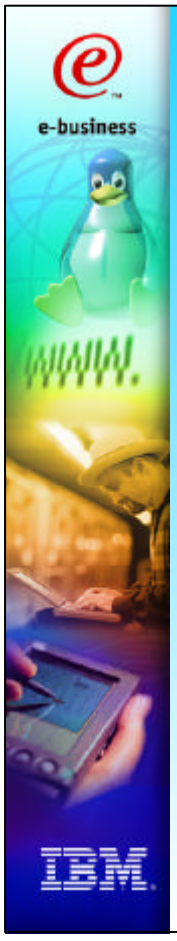
- **Service/Support Engineer's Facility**
  - Live Systems
  - Non-recreatable Problems
  - No source modification required
  - Timing Sensitive Problems
- **Developer's Tool**
  - Alternative to temporary printk/printf
  - Application, Driver, Kernel etc.
  - Timing Sensitive Problems
- **Test Tool**
  - Fault Injection



1.3 Dynamic Probes - Mechanics

- **Global Breakpoint Probes**
  - In-core code modification
  - Track by Inode-Offset
  - Avoid COW page privatization using physical address
  - Unlimited Concurrent Probes
- **Global Watchpoint Probes**
  - Uses Debug Registers
  - Track by Virtual Address
- **Pre-probe Script Driven Probe Handler**
  - RPN assembler language interpreter
  - HLL C-like Compiler

Kernel Hooks (GKHI)

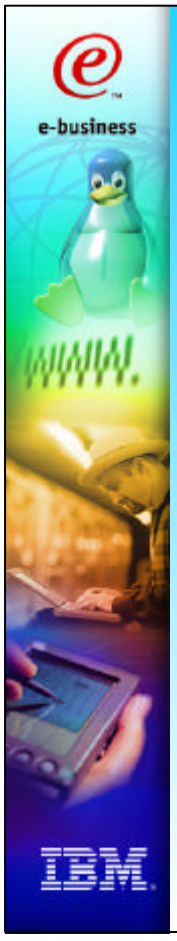


2.1 Kernel Hooks - What are they?

- Code locations where added function may be inserted
- Supplement or replace standard function - subclassing
- Function may not be known at build or run time
- Function may load later therefore simple call cannot be used
- Kernel has a particular need to implement hooks
- Used by DProbes

2.2 Why not Patch?

- **Inconvenient**
  - **Multiple patches may require manual rework**
- **Inflexible**
  - **Cannot select additional functions at run-time**
- **Code Bloat**
  - **Additional functions always present**
  - **Obscures the prime function**



2.3 Basic Requirements

- **Multiple hooks to co-exist within a module**
- **Shared use of a hook by multiple exits**
- **Sole use of a hook by a specific exit**
- **Minimal impact to performance when a hook is unused**
- **Exit must be able to operate as if inserted:**
  - **Have access to local variables**
  - **Terminate the function**
- **Group of exits able to insert atomically**

- **Need a Managed Interface**

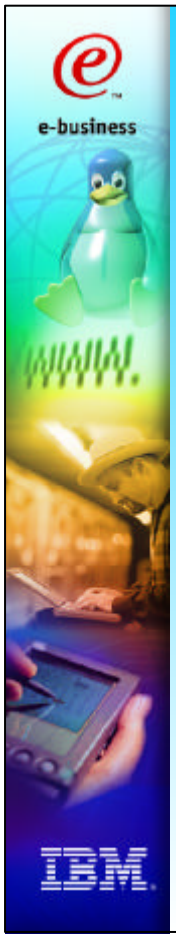
2.4 The Managed Interface

- For Hooked Code:
  - A HOOK macro - indicate the hook location
  - hook_initialise - allows use of the hook
  - hook_terminate - disallows use of the hook
- For Hook Exits:
  - hook_register - identifies exit routine and priority
  - hook_arm - activates group of exits
  - hook_disarm - deactivate group of exits
  - hook_deregister - removes exit from interface

Linux Event Logging for the Enterprise (evlog)

3.1 evlog - What is it?

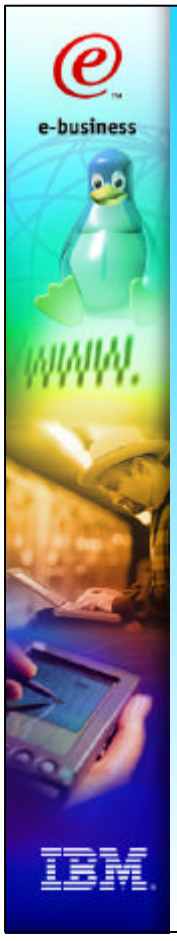
- **Comprehensive Logging Capability**
  - Complies with draft POSIX SRASS standard
  - POSIX APIs
  - Structured Event Records
  - Optionally Captures Syslog and Klog messages
  - Logs Binary and Text Messages
  - User and Kernel Space
  - Task, Init & Interrupt Time
  - Event Notification - Automation, System Management
  - Event Filtering
  - Log Management
  - After-the-fact Formatting



3.2 evlog - Where Used

- **Device Driver Hardening**
  - Automated Recovery
  - On-line Diagnostic Action
  - System Management
- **Instrumentation Schemes**
  - Wrapper macros
  - Ease of Implementation

Flexible Dump



4.1 Flexible Dump - What is it?

- **Goals for a Comprehensive System Dump**
  - **Non-disruptive - Snapshot Capable**
  - **System and (multiple) User Context Visibility**
  - **Minimal System Dependence**
  - **Stand-alone Capable**
  - **Customisable Dumping - Virtual & Physical Memory Ranges, Objects, Processor Resources etc.**
  - **Multiple triggers: Exception Kernel/User, API, NMI/KBD Interrupt**
  - **Access to Swapped Data**
  - **Dump Space/Repository Management**
  - **Programmable formatter**
  - **SMP Capable**
  - **Support for Alternative Dump Devices (Telco)**

4.1 Flexible Dump - Where is it?

- **Contributions to LKCD**
  - **Snapshot Dump - DProbes interface**
  - **Non-disruptive**
  - **Custom Dump Objects**
  - **Minimal System Dependence**
  - **SMP fixes + multiple CPU status saving**
- **Working with LKCD Community**

System Trace

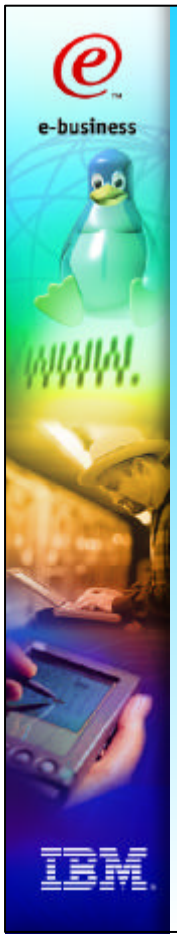
5.1 System Trace - What is it?

- **Generic Trace Recording Mechanism**
- **Community contributions to:**
  - **Linux Trace Toolkit (Opersys)**
    Dynamic Trace - DProbes interface
    Formatting exit for RAW trace data
- **Supporting Similar efforts in:**
  - **Linux Kernel State Trace (LKST) - Hitachi**

Community Adoption

6.1 Adoption Initiatives

- Establishing a RAS Community - OLS RAS BoFs
- Minimise Fragmentation - Maximise Contribution
- Canvassing Distributors
- POSIX
- Instrumentation - standards, aids, implementation
- Porting & Currency

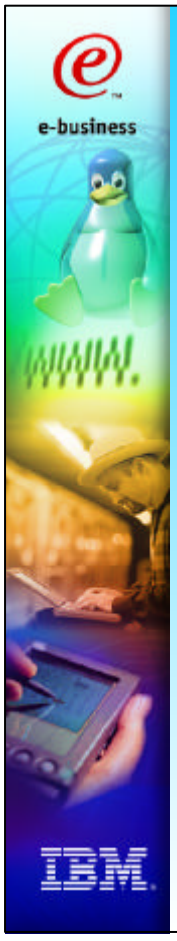


7 Miscellaneous

- KDB
  - ➡ Complex Breakpoints - DProbes Interface
  - ➡ Two Patches Accepted
- Kernel
  - ➡ Debug Register Allocation Patch (Dprobes/KDB/gdb)

8 What's Next?

- Log/Trace Instrumentation of Kernel and Device Drivers
  - ➙ We need participation from the Community
- DProbes ports for IA64 and zSeries
- Turbo Linux release of RAS Utilities
- Sampler Probe type for Profiling
- DProbes HLL Compiler
- Dump User Contexts
- KDB User Contexts
- Mission Critical mcore Integration with LKCD
- On-line Diagnostics Framework
- First Failure System Technology
- Performance Co-Pilot
- RAS Community BOF at OLS



9 The Team - Contacts

Mailing List: *systemras-developers@lists.sourceforge.net*
Web Page: *http://systemras.sourceforge.net/*
Richard J Moore: *richardj_moore@uk.ibm.com*

India:

Suparna Bhattacharya
S. Vamsikrishna
Subodh Soni
Bharata B. Rao

UK:

Richard J Moore (Architect)

USA:

Larry Kessler
James Keniston
Haren Myneni
Hien Q Nguyen
Mike Sullivan
Michael Mason
Thomas Zanussi
Daniel Stekloff
David Oleszkiewicz
Tom Hanrahan (Manager)

End of Presentation

Trademarks

- IBM, zSeries and S/390 are trademarks of the International Business Machines Corporation in the United States and other countries.
- IA32 and IA64 are abbreviations Pentium 32-bit and Itanium 64-bit architectures of the Intel Corporation.