

Unified Kernel Bug Collection

John Harrison

October 2026



Bugs Are Hard!



Bug reporting is hard

- Especially so for end users!
- Information is in lots of places
 - `dmesg`
 - `/var/log/syslog`
 - `/sys/class/drm/card%d/error`
 - `/sys/class/devcoredump/devcd<m>`
 - `/var/crash`
 - ...
- Users don't know where or how to file a bug
- Users might not even know that a crash has happened
 - Many errors are recoverable
 - User just sees a brief stutter or glitch and either misses it or ignores it)
 - Or a server process crashes but auto-restarts and no-one sees anything
- So most bug instances are probably never reported at all



Bug triage is hard

- Is it really a one-off, never to be seen again?
- Were all the important logs included?
 - Were the logs only saved two hours after the bug occurred and have wrapped?
- Bugs can cascade
 - a non-fatal issue can sometimes lead to a fatal issue later in time
 - need a timeline of reports from a single system/boot
- Bugs can cross subsystem boundaries
 - the crash might be in driver X but was really caused by module Y
 - if bug is filed against X then Y might never know about it



Solution:

Remove the user!



Automated Bug Reports

1. Automatically generate a dump file with all relevant kernel information
2. Send notification to user side daemon
3. Daemon collects all useful non-kernel information
4. Daemon sends collected bug report to global server
5. Server does first step triage
6. Server provides database for investigating bugs



1. Kernel Data Collection

- Simply spamming to dmesg is insufficient for many issues
- Need to include all relevant state - driver internals, hardware registers, memory buffers, generated microcode programs, etc.
- Need to capture all that state at the point of the bug
 - devcoredump is one good mechanism for this
 - allows capture of arbitrary data
 - encapsulated into a single entity
 - can be huge - many gigabytes even
 - can be made both human readable and machine readable
 - downside is no common framework/spec for content



2. Notification to Userland

- Kernel sends a notification to a user side daemon
 - e.g. via uevent



3. Non-kernel Data Collection

- Daemon retrieves the kernel dump file
- Daemon collects all useful system information:
 - the kernel dump file (devcoredump or equivalent)
 - unique session id (e.g. hash of processor id, system boot time, etc.)
 - dmesg
 - syslog
- Potential for application hooks to include more information about the task that died (e.g. mesa, OpenCL, etc. libraries hook in and provide internal state)
- Need to be careful about collecting sensitive information – provide hooks/filters to allow redaction



4. Report Externally

- Daemon sends collected bug report to global server
- Biggest problem – who runs the servers?



5. First Level Triage

- Build timeline of bug reports with the same session id
 - Check if new bug report occurred after a known instability issue and can be assigned to the earlier bug instance instead
- Match new bug to existing reports
- Run driver/module specific hooks on relevant bugs for customised triage
 - e.g. decoding devcoredump blob to extract basic info about the bug
- Send notification to driver/module owner when a new bug is identified



6. Developer Database

- Developers can browse all the bug reports related to their area
- Can see how many times a given bug has really occurred
- Can see what platforms/hardware/kernels/etc. are affected
- Can see all the important logs/data for every bug instance
- Can easily re-assign a bug to a different area as required



Making It Happen



Making It Happen

- Need someone to host a unified bug server that everyone, everywhere can connect to
- Need to establish a common bug file format that can contain all relevant data / metadata
 - Ideally human readable but still containing arbitrary binary data
- Need to implement the collection daemon
 - Need to provide options for limiting collection of sensitive data
 - e.g. including the OpenCL code that crashed a GPU would be useful for debugging but could expose a company's proprietary algorithms and other IP
- Need all distros to run the collection daemon by default
- Need to implement the bug server itself
- Bug server can be an existing entity (bugzilla, jira, etc.) but will need some newly defined interface for submitting kernel bugs using above file format, extracting useful info, doing first level triage, etc. in order to create a bug within the database backend



The End

Join us!

<https://www.igalia.com/jobs>



