A bug is NOT a bug is NOT a bug

Differences in bug classes, bug tracking and bug impact
What's a bug?

A system call changing the errno values it returns, although the "specification" claims system call behaviour remains backwards-compatible.

Spelling mistake in a comment

A static analysis finding that should be addressed

Compiler warning

Code snippets in Documentation that do not work

A failing test case in xfs_tests

A missing error check

Unintentional dead code

Race Conditions

Information leaks from the kernel

A system call changing the errno values it returns, although the “specification” claims system call behaviour remains backwards-compatible.
Bug (in the widest sense we can think of)

Every commit that does not add a feature is a bug-fix commit.
... and the initial trigger that motivates the change is the *bug*.

**Definition:**
A Bug is one locally-bound aspect of the content in the kernel repository
- that motivates a change to the kernel repository
- without adding significant new functionality for any stakeholder
The Kernel’s State of the Union

- Who identifies bugs?
- How are bugs reported?
- How are bugs tracked?
- Who cares about specific bugs?
- Which actions happen due to a bug report?
  - How fast? How reliable?
Bug Detector
- Finds bug on various versions, configs
- Checks fix

Bug Tracker
- tracks status
- relates similar reports to each other
- aggregates information

Upstream Development Community (Maintainers, Bug-fixing developers)

Users & Downstream Community
Decides when to update products?

Provides fix, request check

Update status, provide related info

Summary of currently known bugs

Update status

Reports bugs
Examples of “bug tracking systems” in the kernel

Let’s look at bug tracking for selective bug classes:

- **Bug report** by a human kernel user
- **Bug report** by an automated testing system (a dedicated repeatable test)
- Some “compiler warning” **bug report**
- **Bug report** identified by fuzzing
- Some “static analysis” **bug report**
Bug report by a human kernel user

Current resources:

  reworked by Thorsten Leemhuis
- bugzilla.kernel.org, regressions@lists.linux.dev
- linux distro bug trackers

Potential future tools:

- regzbot (https://linux-regtracking.leemhuis.info/post/regzbot-approach/)
  envisioned by Thorsten Leemhuis
Bug report by an automated testing system

Current resources:

- Lava, Fuego, Kernel CI (and kcldb)...

But… How does bug tracking work?

- What is suitable identifier for a bug reported by a failed test suite?
- How is bug and its fix linked to a change from non-failing to failing test and vice versa?
- How to link a commit fix to a specific interval or collection (e.g., on different configs) of failing tests?
Some “compiler warning” *bug report*

Current resources:

- Stephen Rothwell's email notifications on linux-next mailing lists
- ClangBuiltLinux's linux github tracker
- Kernel CI build logs
- kernel test robot (Intel's 0-day testing)

*Precise parsing? Automation? Collaborative tracking and reporting?*
Bug report identified by fuzzing

Current resources:

- Syzbot (https://syzkaller.appspot.com/upstream)
  - Comes with its own “bug tracker” machinery
  - Needed to reinvent “bug tracking” because there was nothing suitable available in the community…

- Other syzkaller instances (from other users):
  - How do they report and are we then managing bug tracking with multiple non-synchronized syzbot instances?

Can we provide something suitable before the next tool comes along and builds yet another bug tracker?
Some “static analysis” **bug report**

Current resources:

- Coverity and coverity-bot (Colin King, Kees Cook)
- https://codechecker.elisa.tech (in early testing)
Providing summaries of bug reports to maintainers

All these systems can aggregate information…

… but what do maintainers expect to see to do their job? And when?
Discussion

Which other tools and resources are used? **Please share in the chat!**

What is missing for bug reporters?

… and for bug report consumers?

Different tools are needed… but where is potential for synergies?

How can we connect “bug reporters” and “bug reporter consumers” in a suitable way?

What features are helpful for bug tracker tools?

How can (novice) community members help in “tracking bugs” beyond what tools can do to support developers and maintainers?