

regression tracking

Thorsten Leemhuis

regtracking with regzbot;

since ~Oct 2021 I'm tracking
regression reports again

(I already did it in 2017 for a while, completely manually)



regzbot

Project information

Repository

Issues 4

Merge requests 1

CI/CD

Deployments

Packages & Registries

Monitor

Analytics

Wiki

Snippets

Thorsten Leemhuis > regzbot



regzbot

Project ID: 29206106

Star 1

247 Commits 1 Branch 0 Tags 2.7 MB Project Storage

Topics: linux kernel regression + 1 more

<https://linux-regtracking.leemhuis.info>

main regzbot

Find file

Download

Clone



general: detect bugzilla links properly in commits and mails

Thorsten Leemhuis authored 2 days ago

c669befd



README

GNU AGPLV3

Name	Last commit	Last update
docs	general: add 'duplicate' (short: dup) c...	1 month ago
regzbot	general: detect bugzilla links properly ...	2 days ago
testdata/expected	general: first part of bugzilla support, ...	1 month ago

regtracking with regzbot;

designed to ideally create **no** additional
work for Linux kernel developers

regtracking with regzbot;

obviously someone needs to add
regression reports to the tracking

regtracking with regzbot;

ideally, the reporter does that by including a paragraph like this in a mail CCed to the usual mailing lists (lkml, regressions, ...):

#regzbot introduced v5.19..v6.0-rc1

regtracking with regzbot;

or a paragraph like like this,
if the regression was not yet bisected:

#regzbot introduced **1a2b3c4d5e6f**

regtracking with regzbot;

anyone (often me) can point regzbot to reports from other people in a reply:

#regzbot introduced v5.19..v6.0-rc1 ^

regtracking with regzbot;

regzbot then looks out for replies
to the report and...

regtracking with regzbot;

...patches posted or committed
pointing to the report using a "Link:" tag

regtracking with regzbot;

...patches posted or committed
pointing to the report using a "Link:" tag
and in the latter case considers the regression resolved

regtracking with regzbot;

yes, that's what 'Link:' tags are for, as
Linus recently clarified multiple times

regtracking with regzbot;

yes, that's what 'Link:' tags are for, as
Linus recently clarified multiple times

and FWIW, he also said: no "BugLink" et. al.;
see my slides from my kernel summit talk for links to lore

regtracking with regzbot;

I add regression reports to the tracking
I find on the regressions mailing list,
in lore, and on bugzilla.kernel.org

sadly my search jobs seem to miss quite a few reports :-/

Linux kernel regression status

[\[next\]](#) [\[mainline\]](#) [\[stable/longterm\]](#) [\[dormant\]](#) [\[resolved\]](#) | [\[new\]](#) | [\[all\]](#)

current cycle (v5.19.. aka v6.0-rc), culprit identified

- [88f1669019bd](#) (v6.0-rc1) ▶ **Re: [PATCH v2 2/2] scsi: sd: Rework asynchronous resume support** by [Vlastimil Babka](#), [Mike Galbraith](#), and [Thorsten Leemhuis](#)
Earliest & latest [activity](#): [9](#) & [1](#) days ago. Noteworthy: [\[1\]](#), [\[2\]](#), [\[patch \(SOB\)\]](#).
- [1aa91d9c99](#) (v6.0-rc1) ▶ **[xfstests] 1aa91d9c99: xfstests.generic.471.fail** by [kernel test robot](#)
Earliest & latest [activity](#): 2 days ago.
- [59bb69c67c](#) (v6.0-rc1) ▶ **[copy_page_{to,from}_iter()] 59bb69c67c: hackbench.throughput -37.6% regression** by [kernel test robot](#)
Earliest & latest [activity](#): 3 days ago.
- [5a46079a9645](#) (v6.0-rc1) ▶ **pm: booting on NXP i.MX8ULP broke** by [Peng Fan](#)
Earliest & latest [activity](#): [28](#) & [5](#) days ago. Noteworthy: [\[1\]](#), [\[patch\]](#).
- [26afbd826ee3](#) (v6.0-rc1) ▶ **[bisected][regression] mediatek bluetooth 13d3:3563 (mt7921e) doesn't work with audio devices.** by [Arek Ruśniak](#) and [Arek Ruśniak](#)
Earliest & latest [activity](#): [6](#) & [5](#) days ago.
- [c3e0c8c2e8](#) (v6.0-rc1) ▶ **[KVM] c3e0c8c2e8: leaking-addresses.proc..data..ro_after_init.** by [kernel test robot](#)
Earliest & latest [activity](#): 10 days ago.

current cycle (v5.19.. aka v6.0-rc), unkown culprit

- v5.19..v6.0-rc2 ▶ **acpi wake up with black screen(failed to get iomux index)** by [neoe](#) and [neoe](#)
Earliest & latest [activity](#): [8](#) & [0](#) days ago.
- v5.19..v6.0-rc1 ▶ **pci or amdgpu: Uncorrected errors reported for AMD GPU** by [Tom Seewald](#) and [Bjorn Helgaas](#)
Earliest & latest [activity](#): [7](#) & [0](#) days ago. Noteworthy: [\[1\]](#), [\[patch \(SOB\)\]](#).
- v5.19..v6.0-rc1 ▶ **New 6.1 net/mac80211/rx.c warning with iwlfwifi / Ultimate-N 6300 wifi** by [Hans de Goede](#)
Earliest & latest [activity](#): [5](#) & [3](#) days ago.

previous cycle (v5.18..v5.19), culprit identified, with activity in the past three months

- [cdf0b86b250f](#) ▶ **net: r8152: ethernet port on Lenovo Thunderbolt 3 dock goes crazy** by [Maxim Levitsky](#)

<https://linux-regtracking.leemhuis.info/regzbot/mainline/>

• v5.19..v6.0-rc1

Earliest & latest activity: 10 & 0 days ago.

▼ **pci or amdgpu: Uncorrected errors reported for AMD GPU** by [Tom Seewald](#) and [Bjorn Helgaas](#)

Earliest & latest activity: 9 & 1 days ago. Noteworthy: [1], [patch (SOB)].

[1]: [\[PATCH 1/2\] drm/amdgpu: Move HDP remapping earlier during init](#)

2 days ago, by Lijo Lazar (monitored) [via dup]

Latest patch: [\[PATCH 2/2\] drm/amdgpu: Init VF's HDP flush reg offset early](#)

2 days ago, by Lijo Lazar; signed-off-by present

Earlier patches: [1](#), [2](#), [3](#)

Latest five known activities:

- [Re: \[Bug 216373\] New: Uncorrected errors reported for AMD GPU](#)
1 days ago, by Christian König
- [Re: \[PATCH 1/2\] drm/amdgpu: Move HDP remapping earlier during init](#)
1 days ago, by Bjorn Helgaas [via dup]
- [Re: \[Bug 216373\] New: Uncorrected errors reported for AMD GPU](#)
1 days ago, by Bjorn Helgaas
- [Re: \[PATCH 1/2\] drm/amdgpu: Move HDP remapping earlier during init](#)
1 days ago, by Felix Kuehling [via dup]
- [Re: \[Bug 216373\] New: Uncorrected errors reported for AMD GPU](#)
1 days ago, by Felix Kuehling

Regzbot command history:

- [title: pci or amdgpu: Uncorrected errors reported for AMD GPU](#)
3 days ago, by Thorsten Leemhuis
- [dup: the regression "Uncorrected errors reported for AMD GPU" was marked as duplicate of this](#)
3 days ago, by Thorsten Leemhuis
- [introduced: v5.19..v6.0-rc1 ^ https://bugzilla.kernel.org/show_bug.cgi?id=216373](#)
3 days ago, by Thorsten Leemhuis

When fixing, add this to the commit message to make regzbot notice patch postings and commits to resolve the issue:

Reported-by: Bjorn Helgaas <helgaas@kernel.org>

Link: <https://lore.kernel.org/r/20220818203812.GA2381243@bhelgaas/>

Reported-by: Tom Seewald <tseewald@gmail.com>

Link: https://bugzilla.kernel.org/show_bug.cgi?id=216373

• v5.19..v6.0-rc1

► **New 6.1 net/mac80211/rx.c warning with iwlwifi / Ultimate-N 6300 wifi** by [Hans de Goede](#)

<https://linux-regtracking.leemhuis.info/regzbot/mainline/>

regtracking with regzbot;

if things stall, I'll notice in the regzbot's
web-ui and will try to prod things

regtracking with regzbot;

and if that doesn't help,
I sometimes get Linus involved

I should have done this more often and will likely do so in
the future – likely asking more often for reverts, too



M+ getting_started.md 12.34 KiB



Open in Web IDE



Get started with regzbot

- Get started with regzbot
 - Why and how to make regzbot track a Linux kernel regression
 - How to let regzbot you are fixing a Linux kernel regression it tracks
 - More regzbot features relevant for both reporters and developers
 - Important basics: How to interact with regzbot
 - Make regzbot track an existing report
 - Update properties of a tracked regression
 - change the range or commit that introduced the regression
 - Update the report's title
 - Point regzbot to other places with further details about a regression
 - Link and monitor a related discussion
 - Point to a place with further details, like a bug-tracker
 - Resolve a regression
 - Mark a regression as fixed
 - Duplicates
 - Mark a regression as invalid

Why and how to make regzbot track a Linux kernel regression

When reporting a Linux kernel regression it is in your interest to make [regzbot](#) aware of the issue, as that ensures the report won't accidentally fall



M+ reference.md 12.82 KiB



Open in Web IDE



Reference documentation for regzbot, the Linux kernel regression tracking bot

- Reference documentation for regzbot, the Linux kernel regression tracking bot
 - Basic concept
 - What regzbot does once it's aware of a regression
 - What regzbot does with the gathered data
 - Interacting with regzbot
 - Commands to be sent as a reply to the report
 - commands to make regzbot track a regression
 - commands to update properties of a tracked regression
 - commands to point to related discussion, reports and webpages
 - commands to resolve a regzbot entry
 - commands users and developers normally shouldn't use
 - Commands regzbot accepts everywhere it looks
 - backlinks
 - tag users and developers normally shouldn't use

Note: this document explains regzbot concept and all options; if you want something easier and quicker to consume, head over to ['getting started with regzbot'](#)

Basic concept

Regzbot is a bot watching mailing lists and Git trees to track Linux kernel regression from report to elimination, to ensure none fall through the

regtracking with regzbot;

sadly, regzbot is still young and has many many warts⁽¹⁾ and deficits^(1, 2)

⁽¹⁾ one of the main reasons: I'm not a good programmer

⁽²⁾ one: it's not really useful for subsystem maintainers

regtracking with regzbot;

sadly, it will take a while before regzbot
becomes more self-serving for
subsystem maintainers :-/ sorry!

regtracking with regzbot;

regression tracking is still
a lot of tedious work, but regzbot
makes it a whole lot easier

regtracking with regzbot;

my regression tracking work is also
far from perfect and
on a "best effort basis"

regtracking with regzbot;

but it helps: quite a number of reports
would afaics have fallen through the
cracks or only addressed much later
especially many reports submitted to bugzilla.kernel.org

regtracking with regzbot;

my experiences made me
write two texts on regressions
for the kernel's docs

Greg ACKed them, Linus applies them without comment

Reporting regressions

The important bits (aka “TL;DR”)

All the details on Linux kernel regressions relevant for users

Security bugs

Bug hunting

Bisecting a bug

Tainted kernels

Ramoops oops/panic logger

Dynamic debug

Explaining the “No working init found.” boot hang message

Documentation for Kdump - The kexec-based Crash Dumping Solution

Performance monitor support

pstore block oops/panic logger

Rules on how to access information in sysfs

ACPI Support

ATA over Ethernet (AoE)

Auxiliary Display Support

» [The Linux kernel user's and administrator's guide](#) » [Reporting regressions](#) [View page source](#)

Reporting regressions

“*We don't cause regressions*” is the first rule of Linux kernel development; Linux founder and lead developer Linus Torvalds established it himself and ensures it's obeyed.

This document describes what the rule means for users and how the Linux kernel's development model ensures to address all reported regressions; aspects relevant for kernel developers are left to [Handling regressions](#).

The important bits (aka “TL;DR”)

1. It's a regression if something running fine with one Linux kernel works worse or not at all with a newer version. Note, the newer kernel has to be compiled using a similar configuration; the detailed explanations below describes this and other fine print in more detail.
2. Report your issue as outlined in [Reporting issues](#), it already covers all aspects important for regressions and repeated below for convenience. Two of them are important: start your report's subject with “[REGRESSION]” and CC or forward it to [the regression mailing list \(regressions@lists.linux.dev\)](#).
3. Optional, but recommended: when sending or forwarding your report, make the Linux kernel regression tracking bot “[reazbot](#)” track the issue by specifying when the regression started like

☐ Handling regressions

The important bits (aka “The TL;DR”)

☒ All the details on Linux kernel regressions relevant for developers

Programming Language

Linux kernel coding style

Subsystem and maintainer tree specific development process notes

Kernel Maintainer PGP guide

Email clients info for Linux

Linux Kernel Enforcement Statement

Kernel Driver Statement

Minimal requirements to compile the Kernel

The Linux Kernel Driver Interface

Linux kernel management style

Everything you ever wanted to know about Linux -stable releases

Linux Kernel patch submission checklist

Index of Further Kernel Documentation

🏠 » [Working with the kernel development community](#) » [Handling regressions](#) [View page source](#)

Handling regressions

We don't cause regressions – this document describes what this “first rule of Linux kernel development” means in practice for developers. It complements [Reporting regressions](#), which covers the topic from a user's point of view; if you never read that text, go and at least skim over it before continuing here.

The important bits (aka “The TL;DR”)

1. Ensure subscribers of the [regression mailing list](#) (regressions@lists.linux.dev) quickly become aware of any new regression report:
 - When receiving a mailed report that did not CC the list, bring it into the loop by immediately sending at least a brief “Reply-all” with the list CCed.
 - Forward or bounce any reports submitted in bug trackers to the list.
2. Make the Linux kernel regression tracking bot “regzbot” track the issue (this is optional, but recommended):
 - For mailed reports, check if the reporter included a line like `#regzbot introduced v5.13. .v5.14-rc1`. If not, send a reply (with the regressions list in CC) containing a paragraph like the following, which tells regzbot when the issue started to happen:

regtracking with regzbot;

the latter reminds developers to

- cc the regression list regressions@lists.linux.dev

regtracking with regzbot;

the latter reminds developers to

- cc the regression list regressions@lists.linux.dev – ideally while telling regzbot about the report in parallel using '#regzbot introduced v5.19..v6.0-rc1^'

regtracking with regzbot;

the latter reminds developers to

- cc the regression list regressions@lists.linux.dev – ideally while telling regzbot about the report in parallel using '#regzbot introduced v5.19..v6.0-rc1^'
- point to the report using 'Link:' tags

regtracking with regzbot;

the latter reminds developers to

- cc the regression list regressions@lists.linux.dev – ideally while telling regzbot about the report in parallel using '#regzbot introduced v5.19..v6.0-rc1^'
- point to the report using 'Link:' tags
- fix regressions quickly, as outlined by a few "rules of thumb"

mainline version.

- Fix regressions **within two or three days**, if they are critical for some reason – for example, if the issue is likely to affect many users of the kernel series in question on all or certain architectures. Note, this includes mainline, as issues like compile errors otherwise might prevent many testers or continuous integration systems from testing the series.
- Aim to fix regressions within **one week** after the culprit was identified, if the issue was introduced in either:
 - a recent stable/longterm release
 - the development cycle of the latest proper mainline release

In the latter case (say Linux v5.14), try to address regressions even quicker, if the stable series for the predecessor (v5.13) will be abandoned soon or already was stamped “End-of-Life” (EOL) – this usually happens about three to four weeks after a new mainline release.

- Try to fix all other regressions within **two weeks** after the culprit was found. **Two or three additional weeks are acceptable for performance regressions and other issues which are annoying**, but don't prevent anyone from running Linux (unless it's an issue in the current development cycle, as those should ideally be addressed before the release). A few weeks in total are acceptable if a regression can only be fixed with a risky change and at the same time is affecting only a few users; as much time is also okay if the regression is already present in the second newest longterm kernel series.

Note: **The aforementioned time frames** for resolving regressions are meant to include getting the fix tested, reviewed, **and merged into mainline**, ideally with the fix being in linux-next at least briefly. This leads to delays you need to account for.

Subsystem maintainers are expected to assist in reaching those periods by doing timely reviews and quick handling of accepted patches. They thus might have to send `git-pull` requests earlier or more often than usual: depending on the fix, it might even be

<https://docs.kernel.org/process/handling-regressions.html> and
Documentation/process/handling-regressions.rst.txt

regtracking with regzbot;

short version: often it's a week or two;

regtracking with regzbot;

short version: often it's a week or two;
for a few regressions more time is okay

regtracking with regzbot;

short version: often it's a week or two;
for a few regressions more time is okay
for others is should just take days

regtracking with regzbot;

short version: often it's a week or two;
for a few regressions more time is okay
for others is should just take days
till the fix is merged in mainline!

regtracking with regzbot;

this concludes the brief overview - see yesterday's talk and its slides for a slightly more detailed description of all of that

regtracking with regzbot;

and thx to Chris and Meta for sponsoring
my efforts these days that started with
funding from the EU through NGI pointer

regtracking with regzbot;

which brings us to the discussion phase

(1) any questions, remarks, or
things to discuss on what
I outlined so far?

regzbot approach, expected time frames for fixing...

(2) quite some regressions take
a long time to get fixed

☐ Handling regressions

The important bits (aka “The TL;DR”)

☒ All the details on Linux kernel regressions relevant for developers

Programming Language

Linux kernel coding style

Subsystem and maintainer tree specific development process notes

Kernel Maintainer PGP guide

Email clients info for Linux

Linux Kernel Enforcement Statement

Kernel Driver Statement

Minimal requirements to compile the Kernel

The Linux Kernel Driver Interface

Linux kernel management style

Everything you ever wanted to know about Linux -stable releases

Linux Kernel patch submission checklist

Index of Further Kernel Documentation

🏠 » [Working with the kernel development community](#) » Handling regressions [View page source](#)

Handling regressions

We don't cause regressions – this document describes what this “first rule of Linux kernel development” means in practice for developers. It complements [Reporting regressions](#), which covers the topic from a user's point of view; if you never read that text, go and at least skim over it before continuing here.

The important bits (aka “The TL;DR”)

1. Ensure subscribers of the [regression mailing list](#) (regressions@lists.linux.dev) quickly become aware of any new regression report:
 - When receiving a mailed report that did not CC the list, bring it into the loop by immediately sending at least a brief “Reply-all” with the list CCed.
 - Forward or bounce any reports submitted in bug trackers to the list.
2. Make the Linux kernel regression tracking bot “regzbot” track the issue (this is optional, but recommended):
 - For mailed reports, check if the reporter included a line like `#regzbot introduced v5.13 . . v5.14-rc1`. If not, send a reply (with the regressions list in CC) containing a paragraph like the following, which tells regzbot when the issue started to happen:

mainline version.

- Fix regressions **within two or three days**, if they are critical for some reason – for example, if the issue is likely to affect many users of the kernel series in question on all or certain architectures. Note, this includes mainline, as issues like compile errors otherwise might prevent many testers or continuous integration systems from testing the series.
- Aim to fix regressions within **one week** after the culprit was identified, if the issue was introduced in either:
 - a recent stable/longterm release
 - the development cycle of the latest proper mainline release

In the latter case (say Linux v5.14), try to address regressions even quicker, if the stable series for the predecessor (v5.13) will be abandoned soon or already was stamped “End-of-Life” (EOL) – this usually happens about three to four weeks after a new mainline release.

- Try to fix all other regressions within **two weeks** after the culprit was found. **Two or three additional weeks are acceptable for performance regressions and other issues which are annoying**, but don't prevent anyone from running Linux (unless it's an issue in the current development cycle, as those should ideally be addressed before the release). A few weeks in total are acceptable if a regression can only be fixed with a risky change and at the same time is affecting only a few users; as much time is also okay if the regression is already present in the second newest longterm kernel series.

Note: **The aforementioned time frames** for resolving regressions are meant to include getting the fix tested, reviewed, **and merged into mainline**, ideally with the fix being in linux-next at least briefly. This leads to delays you need to account for.

Subsystem maintainers are expected to assist in reaching those periods by doing timely reviews and quick handling of accepted patches. They thus might have to send `git-pull` requests earlier or more often than usual: depending on the fix, it might even be

<https://docs.kernel.org/process/handling-regressions.html> and
Documentation/process/handling-regressions.rst.txt

sometimes sluggish fixing;

sadly, most regressions take
way longer to get fixed :-/

sometimes sluggish fixing;

problem areas outlined in yesterdays talk
and covered by handling-regressions.rst:

sometimes sluggish fixing;

(a) due to various reasons, regression fixes take weeks till they are sent to Linus

sometimes sluggish fixing;

one of the reasons:

sometimes even simple fixes hang around
in -next till the next merge window⁽¹⁾

(1) which contributes to the big stable updates right after rc1 :-/

sometimes sluggish fixing;

one of the reasons:

developers don't sent urgent fixes directly upstream (e.g. to Linus) when a subsys. maintainers is MIA for some reason

sometimes sluggish fixing;

(b) regressions introduced in earlier mainline cycles seem to be a kind of second-class citizen for some developers and maintainers

sometimes sluggish fixing;

I'm working against these and
a few other aspects and
things got slightly better already
takes a while to spread the word...

sometimes sluggish fixing;

two main remaining problems:

1. developers try hard to avoid reverts
2. reviewers and maintainers sometimes miss when a patch fixes a regression

sometimes sluggish fixing;

two main remaining problems:

- 1. developers try hard to avoid reverts**
2. reviewers and maintainers sometimes miss when a patch fixes a regression

sometimes sluggish fixing;

I wonder if we could improve things
by making reverts less feared
(aka more attractive)

sometimes sluggish fixing;

785538bfdd68 ("scsi: sd: Revert "Rework asynchronous resume support")

sometimes sluggish fixing;

caused suspend to fail on a lot of
machines and many common laptops

sometimes sluggish fixing;

2022-07-07 "scsi: sd: Rework asynchronous resume support" committed as 88f1669019bd

2022-07-19 first bisected regression report while in linux-next [1]

2022-08-04 merged to mainline via SCSI subsystem

2022-08-14 Linux 6.0-rc1 is out

2022-08-16 revert posted for review [2]

2022-08-16 backport to stable proposed, but prevented in time [3]

2022-08-16 second bisected regression report [4]

2022-08-17 third bisected regression report [5]

2022-08-17 fourth bisected regression report [6]

2022-08-19 report about issues caused by the commit [7]

2022-08-21 fifth bisected regression report [8]

2022-08-22 sixth bisected regression report [9]

2022-08-22 Vbabka asking for a promptly revert (no reply) [10]

2022-08-25 seventh bisected regression report [11]

2022-08-26 revert in mainline as 785538bfdd68 [12]

2022-08-28 Linux 6.0-rc3 is released

sometimes sluggish fixing;

References:

[1] <https://lore.kernel.org/linux-scsi/alpine.DEB.2.22.394.2207191125130.1006766@ramsan.of.borg/>

[2] <https://lore.kernel.org/linux-scsi/20220816172638.538734-1-bvanassche@acm.org/>

[3] <https://lore.kernel.org/all/b532e50f-7aa0-5ac3-c7a6-6a43ab9c1bc9@acm.org/>

[4] https://bugzilla.kernel.org/show_bug.cgi?id=215880#c31

[5] <https://lore.kernel.org/linux-scsi/8a83665a-1951-a326-f930-8fcbb0c4dd9a@huawei.com/>

[6] <https://lore.kernel.org/lkml/98592410-dd31-9081-86be-fda67d3b06d2@suse.cz/>

[7] <https://lore.kernel.org/regressions/dd6844e7-f338-a4e9-2dad-0960e25b2ca1@redhat.com/>

[8] <https://lore.kernel.org/all/ca8052efe4d1357bc6ece0a45e8429de37e3ae03.camel@gmx.de/>

[9] https://bugzilla.kernel.org/show_bug.cgi?id=216398

[10] <https://lore.kernel.org/regressions/f7aad839-2116-ab85-8ad5-e8d2f7b10c43@suse.cz/>

[11] https://bugzilla.kernel.org/show_bug.cgi?id=216413

[12] <https://git.kernel.org/torvalds/c/785538bfdd682c8e962341d585f9b88262a0475ez>

sometimes sluggish fixing;

2022-07-07 "scsi: sd: Rework asynchronous resume support" committed as 88f1669019bd

2022-07-19 first bisected regression report while in linux-next [1]

2022-08-04 merged to mainline via SCSI subsystem

2022-08-14 Linux 6.0-rc1 is out

2022-08-16 revert posted for review [2]

2022-08-16 backport to stable proposed, but prevented in time[3]

2022-08-16 second bisected regression report[4]

2022-08-17 third bisected regression report[5]

2022-08-17 fourth bisected regression report[6]

2022-08-19 report about issues caused by the commit [7]

2022-08-21 fifth bisected regression report[8]

2022-08-22 sixth bisected regression report[9]

2022-08-22 Vbabka asking for a promptly revert (no reply) [10]

2022-08-25 seventh bisected regression report[11]

2022-08-26 revert in mainline as 785538bfdd68 [12]

2022-08-28 Linux 6.0-rc3 is released

sometimes sluggish fixing;

2022-07-07 "scsi: sd: Rework asynchronous resume support" committed as 88f1669019bd

2022-07-19 first bisected regression report while in linux-next [1]

2022-08-04 merged to mainline via SCSI subsystem

2022-08-14 Linux 6.0-rc1 is out

2022-08-16 revert posted for review [2]

2022-08-16 backport to stable proposed, but prevented in time[3]

2022-08-16 second bisected regression report[4]

2022-08-17 third bisected regression report[5]

2022-08-17 fourth bisected regression report[6]

2022-08-19 report about issues caused by the commit [7]

2022-08-21 fifth bisected regression report[8]

2022-08-22 sixth bisected regression report[9]

2022-08-22 Vbabka asking for a promptly revert (no reply) [10]

2022-08-25 seventh bisected regression report[11]

2022-08-26 revert in mainline as 785538bfdd68 [12]

2022-08-28 Linux 6.0-rc3 is released

sometimes sluggish fixing;

2022-07-07 "scsi: sd: Rework asynchronous resume support" committed as 88f1669019bd

2022-07-19 first bisected regression report while in linux-next [1]

2022-08-04 merged to mainline via SCSI subsystem

2022-08-14 Linux 6.0-rc1 is out

2022-08-16 revert posted for review [2]

2022-08-16 backport to stable proposed, but prevented in time[3]

2022-08-16 second bisected regression report[4]

2022-08-17 third bisected regression report[5]

2022-08-17 fourth bisected regression report[6]

2022-08-19 report about issues caused by the commit [7]

2022-08-21 fifth bisected regression report[8]

2022-08-22 sixth bisected regression report[9]

2022-08-22 Vbabka asking for a promptly revert (no reply) [10]

2022-08-25 seventh bisected regression report[11]

2022-08-26 revert in mainline as 785538bfd68 [12]

2022-08-28 Linux 6.0-rc3 is released

sometimes sluggish fixing;

2022-07-07 "scsi: sd: Rework asynchronous resume support" committed as 88f1669019bd

2022-07-19 first bisected regression report while in linux-next [1]

2022-08-04 merged to mainline via SCSI subsystem

2022-08-14 Linux 6.0-rc1 is out

2022-08-16 revert posted for review [2]

2022-08-16 backport to stable proposed, but prevented in time[3]

2022-08-16 second bisected regression report[4]

2022-08-17 third bisected regression report[5]

2022-08-17 fourth bisected regression report[6]

2022-08-19 report about issues caused by the commit [7]

2022-08-21 fifth bisected regression report[8]

2022-08-22 sixth bisected regression report[9]

2022-08-22 Vbabka asking for a promptly revert (no reply) [10]

2022-08-25 seventh bisected regression report[11]

2022-08-26 revert in mainline as 785538bfdd68 [12]

2022-08-28 Linux 6.0-rc3 is released

sometimes sluggish fixing;

consider the situation partly my fault,
should have pointed James and Linus to
this earlier and asked for a quick revert

will do that more often in the future,
even if that will make me the bad guy

sometimes sluggish fixing;

real cause: the developer tried
hard avoiding a revert for too long

not unusual, as most developers are happy if their code
finally landed in mainline and try to avoid a big setback

sometimes sluggish fixing;

ease this by allowing reapplying reverted culprits to mainline together with a fix within ~two weeks of the revert?

e.g. outside of the merge windows up to maybe -rc5 for any not-crucial change?

sometimes sluggish fixing;

even if the culprit was merged in an earlier cycle, as reverts then are even more feared upon there

sometimes sluggish fixing;

that would also help if reviewing
a proper fix takes some time when
the culprit is important, too
esp. if the issue was backported to stable

sometimes sluggish fixing;

5467801f1fcb ("gpio: Request interrupts
after IRQ is initialized")

fixes a NULL pointer dereference

maintainer (Bartosz) hasn't picked it up to send to you.

It's a severe problem; anyone who hits it:

- 1) Power button doesn't work anymore
- 2) Can't resume their laptop from S3 or s2idle

Because the original patch was cc stable@ it landed in stable releases

maintainer (Bartosz) hasn't picked it up to send to you.

It's **a severe problem**; anyone who hits it:

- 1) Power button doesn't work anymore
- 2) Can't resume their laptop from S3 or s2idle

Because the **original patch was cc stable@**, it landed in stable releases and has been **breaking people left and right as distros track the stable channels**. The patch is well tested. Would you please consider to pick this up directly to fix that regression?

Thanks,

Mario Limonciello (1):

gpio: Request interrupts after IRQ is initialized

drivers/gpio/gpiolib.c | 4 ++--

1 file changed, 2 insertions(+), 2 deletions(-)

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

References:

[1] <https://lore.kernel.org/lkml/BL1PR12MB51577A77F000A008AA694675E2EF9@BL1PR12MB5157.namprd12.prod.outlook.com/>

[2] <https://lore.kernel.org/linux-gpio/20220414025705.598-1-mario.limonciello@amd.com/>

[3] <https://gitlab.freedesktop.org/drm/amd/-/issues/1976>

[4] https://www.reddit.com/r/Dell/comments/u5hajd/psa_linux_5173_on_dell_amd_laptops_might_cause/

[5] <https://lore.kernel.org/linux-gpio/e0c79586-3501-050d-f279-2506770324ee@leemhuis.info/>

[6] https://bugzilla.kernel.org/show_bug.cgi?id=215850

[7] <https://gitlab.freedesktop.org/drm/amd/-/issues/1979>

[8] https://bugzilla.suse.com/show_bug.cgi?id=1198697

[9] <https://lore.kernel.org/linux-gpio/de25abef-c071-9f71-36dd-8f2f0b77dc28@leemhuis.info/>

[10] <https://lore.kernel.org/linux-gpio/ae465387-7d77-a208-2c9d-18d6ffad69a0@leemhuis.info/>

[11] <https://lore.kernel.org/all/20220422131452.20757-1-mario.limonciello@amd.com/>

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

2022-04-10 5467801f1fcb ("gpio: Request interrupts after IRQ is initialized") merge to mainline

2022-04-11 Linux 5.18-rc2 released

2022-04-12 backport of 5467801f1fcb part of the rc1 releases of 5.10.111, 5.15.34, 5.16.20, and 5.17.3

2022-04-13 5.10.111, 5.15.34, 5.16.20, and 5.17.3 released with a backport of 5467801f1fcb

2022-04-14 Mario Limonciello reports the regression [1]

2022-04-14 Mario proposes a fix [2]

2022-04-16 another report [3]

2022-04-17 regression makes the news [4]

2022-04-18 /me asking for a revert in the stable tree [5]

2022-04-18 another report [6]

2022-04-19 another report [7]

2022-04-20 another report [8]

2022-04-21 /me prodding [9]

2022-04-22 /me prodding again, asking Mario to submit directly to Linus [10]

2022-04-22 Mario submits directly to Linus [11]

2022-04-22 06fb4ecfeac7 ("gpio: Request interrupts after IRQ is initialized") merged to mainline

2022-04-24 Linux 5.18-rc4 released

2022-04-25 subsystem maintainer notices that Mario's patch was in his spam folder

2022-04-27 5.10.113, 5.15.34, and 5.17.5 released with a backport of 06fb4ecfeac7

sometimes sluggish fixing;

(2.1) fix regressions faster

- allow reapplying reverted regression culprits to mainline once a fix is ready⁽¹⁾?

⁽¹⁾ outside of the merge window, within reasonable limits, e.g. not directly before a mainline release and only if the revert was not too long ago

sometimes sluggish fixing;

two main remaining problems:

1. developers try hard to avoid reverts
- 2. reviewers and maintainers sometimes miss when a patch fixes a regression**

sometimes sluggish fixing;

would a special tag help, or
should we simply teach developers to
write better patch descriptions?

sometimes sluggish fixing;

'Label: #regressionfix'

sometimes sluggish fixing;

[PATCH] foo: bar: Fix odd corner case when profile support is used

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Label: #regressionfix

Fixes: cfc85f3e4b7f ("pci/bar: Add profile support to something")

Reported-by: Holger Tang <holger@example.com>

Link: <https://lore.kernel.org/r/foo-20222342323423423-23234@example.com>

Signed-off-by: Charles Liang <charles@example.com>

Signed-off-by: Leon Someone <leon@example.com>

[...]

sometimes sluggish fixing;

free-form tag line could be used for other things as well, for example "#nobackport" or "#nostable" ?

(2.2) fix regressions faster

- make it more obvious if a patch is fixing a regression? special tag?
-

(3) bugzilla.kernel.org:
isn't it overdue we do
something about it?

bugzilla.kernel.org

I noticed quite some good⁽¹⁾
regression reports there that
apparently nobody looked at

⁽¹⁾ obviously there are bad ones, too

0 siblings, 2 replies, 0 messages in thread
From: Thorsten Leemhuis @ 2022-04-06 12:35 UTC ([permalink](#) / [raw](#))
To: Linus Torvalds, Greg KH, Konstantin Ryabitsev
Cc: regressions, Linux Kernel Mailing List, workflows

Hi! TLDR: I looked closer at every ticket filed in bugzilla.kernel.org over a time span of two weeks to see how well reports are handled, in particular those for kernel regressions. The results of this rough analysis are kinda devastating from my point of view. I for example found 8 tickets describing a regression where the reporter had even bisected the problem, but nevertheless the ticket afaics didn't get a single reply or any other reaction from a regular kernel developer within about a week; in fact out of a total of 20 reports that looked like regressions to me (17 if you exclude tickets where the reporter used an afaics lightly patched distro kernel), only one got a helpful reply from a developer within a week. That makes us miss valuable reports and puts our "no regressions" rule into a bad light. Hence, something IMHO should be done here to improve the situation, but I'm not sure myself what exactly -- that's why I'm writing this mail. A better warning on bugzilla's frontpage suggesting to report issues by mail

<https://lore.kernel.org/lkml/6808cd17-b48c-657d-de60-ef9d8bfa151e@leemhuis.info/>

From: [Konstantin Ryabitsev](#) @ 2022-04-20 16:32 UTC ([permalink](#) / [raw](#))

To: Thorsten Leemhuis

Cc: Krzysztof Kozłowski, Linus Torvalds, Greg KH, regressions,
Linux Kernel Mailing List, workflows

On Wed, Apr 20, 2022 at 01:57:12PM +0200, Thorsten Leemhuis wrote:

> > I find such Bugzilla useless - the Components are not matching reality,
> > Products look ok except missing really a lot. Does it have proper
> > assigners based on maintainers? Nope. At least not everywhere.

Nobody has stepped up to maintain bugzilla for the past 10 years. Managing components, products, assignees -- that's not the job of the infrastructure team. Linux development is so compartmentalized that cross-subsystem tasks like bug reporting have been thoroughly neglected.

However, I would argue that bugzilla needs fewer components, not more of them. Otherwise people get confused and file bugs against "kernel.org" or whatever happens to be the first entry in the list. For bugzilla to be useful, it needs to have a bugmaster -- and nobody has volunteered thus far. It's not something that members of the LF IT team can do, since none of us are kernel developers.

If someone steps up, I'll be happy to grant them admin rights to manage all the components, etc.

> > All the bug or issue reports I get via email and I think I am not alone

<https://lore.kernel.org/lkml/20220420163223.kz32qomzj3y4hjj5@nitro.local/>

bugzilla.kernel.org

bugzilla.kernel.org – state of things [very brief and rough]

- server and its software are **well maintained Konstantin + team**
- products, components, default assignees, et. al **are heavily outdated, incomplete, wrong, et. at., as nobody really maintains them**
- never really sanctioned as the official place to report kernel bugs: only 20 out of ~2500 entries in MAINTAINERS tell users to file issues there
- a few other developers & subsystems keep a eye on it, too
- some (a lot?) of tickets afaics are not forwarded to any developer
- a lot of reports (even good ones!) in the end **never get a reply from a developer**

bugzilla.kernel.org

I currently look out for regression reports filed in bugzilla.kernel.org and forward those that look somewhat valid

bugzilla.kernel.org

that's a time-consuming task I don't want
to commit myself to permanently⁽¹⁾

⁽¹⁾ I thus might stop at any time...

bugzilla.kernel.org

and I ignore everything that's not a regression⁽¹⁾, but noticed quite some bug reports likely of interest for developers

⁽¹⁾ there are only so many hours in a day...

bugzilla.kernel.org

brought this up yesterday on the summit
and its hallways

bugzilla.kernel.org

nobody was happy about the state of things; a few people wanted to kill it, but some people definitely wanted to keep it
someone later suggested to make it mandatory for all, too

(3) what to do with bugzilla.kernel.org

- new text for the front page explaining things better?
 - force those that want bugzilla to clean up and maintain components, products et. at.? they could create one that makes "no one will look at this" obvious, too
 - decommission bugzilla? or make everyone keep an eye on bugzilla?
 - find minions that keep an eye on things and act as middleman?
 - leave things as they are
 - [insert suggestion of choice]
-

(4) a few minor things,
if there is time and interest

(4.1) how do you want me to
handle reports with slightly
patches kernels?

(4.2) have reports about
practical issues found by
CI testing in regzbot?

(4.3) hang, panic, oops, bug,
warn, and sanitizers: do you
want me to track them, even if
they are no regression?

(5) Am I holding this regression tracking thingy right or do you want me to do something differently?
Anything else regarding regressions?
