

Linux kernel regression tracking

state of the union and discussion on the state of things

Thorsten Leemhuis

intro;

part 1: a really quick overview
of my work on regression tracking

intro;

part 2: some best practices for developers
drawn from my experiences

intro;

part 3: problem areas I'd like to discuss

intro;

note, right after the talk I sadly need to head to OSS EU to give a talk there, but I hope to get back here during lunch in case anyone wants to talk to me

intro;

be warned, I'll rush
through the first two parts!

the slides are kinda self-explanatory and available

regtracking with regzbot

[1. my recent regression
tracking efforts]

regtracking with regzbot

since ~Oct 2021 I'm tracking
regression reports again

regtracking with regzbot

I already did it in 2017 for a while

in my spare time, completely manually, which is a lot of frustrating and labor-intensive work – which is why it gave up in the end



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

tried to not write yet another bug tracker

[but in the end it obviously is one tailored specifically
to the needs of Linux kernel development model]

regtracking with regzbot

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

it also doesn't spam developers!

[at least for now]

regtracking with regzbot

obviously, someone needs to tell
regzbot about regression reports

regtracking with regzbot

ideally, the reporter does that by including a paragraph like this in mailed reports:

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

regtracking with regzbot

or a paragraph like like this,
if the regression was bisected:

```
#regzbot introduced 1a2b3c4d5e6f
```

regtracking with regzbot

someone else (me for example) can point regzbot to reports in a reply, too:

```
#regzbot introduced 1a2b3c4d5e6f ^
```


regtracking with regzbot

adding bugzilla tickets or arbitrary links
by mail is now supported, too

#regzbot introduced 1a2b3c4d5e6f https://bugzilla.kernel.org/show_bug.cgi?id=123456789

regtracking with regzbot

telling regzbot about reports is all the
manual overhead in the ideal case!

regtracking with regzbot

right now it's mostly me that's using this
to add regressions to regzbot's tracking

regtracking with regzbot

I look out for regressions reports on the
regressions mailing list, lore, and
bugzilla.kernel.org

regtracking with regzbot

regzbot then looks out for replies
to tracked reports, but also...

regtracking with regzbot

...for patches posted or committed that point to tracked reports using a "Link:" tag

regtracking with regzbot

...for patches posted or committed that point to tracked reports 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 while emphasizing that he considers them to be important

1 siblings, 0 replies, 59 messages in thread
From: Linus Torvalds @ 2022-05-11 16:20 UTC ([permalink](#) / [raw](#))
To: Michael Ellerman
Cc: Michael S. Tsirkin, Konstantin Ryabitsev, KVM list,
virtualization, Netdev, Linux Kernel Mailing List, mie

On Wed, May 11, 2022 at 3:12 AM Michael Ellerman <mpe@ellerman.id.au> wrote:
>
> Which I read as you endorsing Link: tags :)

I absolutely adore "Link:" tags. They've been great.

But they've been great for links that are **useful**.

They are wonderful when they link to the original problem.

They are **really** wonderful when they link to some long discussion about how to solve the problem.

They are completely useless when they link to "this is the patch submission of the SAME DAMN PATCH THAT THE COMMIT IS".

2022-06-28 8:43 Thomas Zimmermann
0 siblings, 1 reply; 7+ messages in thread

From: Linus Torvalds @ 2022-06-27 17:25 UTC ([permalink](#) / [raw](#))

To: Javier Martinez Canillas

Cc: amd-gfx list, Hans de Goede, dri-devel, Thomas Zimmermann,
Alex Deucher, Christian König, Zack Rusin

On Mon, Jun 27, 2022 at 1:02 AM Javier Martinez Canillas

<javierm@redhat.com> wrote:

>

> The flag was dropped because it was causing drivers that requested their
> memory resource with `pci_request_region()` to fail with `-EBUSY` (e.g: the
> `vmwgfx` driver):

>

> <https://www.spinics.net/lists/dri-devel/msg329672.html>

See, **that** link would have been useful in the commit.

Rather than the useless link it has.

Anyway, removing the busy bit just made things worse.

just wish I could see `_what_` the issue was.

Put another way: I can see that

Reported-by: Zhangfei Gao <zhangfei.gao@foxmail.com>

in the commit, but I don't have a clue what the actual report was, and there really isn't enough information in the commit itself, except for a fairly handwavy "Device drivers might, for instance, still need to flush operations.."

I don't want to know what device drivers `_might_` do. I would want to have an actual pointer to what they do and where.

I suspect it's related to `mmu_notifiers` or something, but I really would have liked a more exact "this is where things go wrong".

I also **suspect** that this is all about that `_loong_` thread (picking one email almost at random) here:

<https://lore.kernel.org/all/a139dbad-2f42-913b-677c-ef35f1eebfed@intel.com>

<https://lore.kernel.org/all/CAHk=-wjMmSZzMJ3Xnskdg4+GGz=5p5p+GSYyFBTh0f-DgvdBWg@mail.gmail.com/>

regtracking with regzbot

"If related discussions or any other background information behind the change can be found on the web, add 'Link:' tags pointing to it. In case your patch fixes a bug, for example, add a tag with a URL referencing the report in the mailing list archives or a bug tracker; [...]"

quote from docs.kernel.org/process/submitting-patches.html
I recently improved that text to make this aspect more clear

2022-06-15 16:28 Petr Mladek
@ 2022-06-15 17:47 Linus Torvalds

-1 siblings, 0 replies; 14+ messages in thread

From: Linus Torvalds @ 2022-06-15 17:47 UTC (permalink / raw)

To: Petr Mladek

Cc: John Ogness, Sergey Senozhatsky, Steven Rostedt,
Paul E . McKenney, Frederic Weisbecker, Peter Geis, Zhouzhouyi,
Davidlohr Bueso, Josh Triplett, rcu, linux-rockchip,
Linux Kernel Mailing List

On Wed, Jun 15, 2022 at 9:28 AM Petr Mladek <pmladek@suse.com> wrote:

>

> BugLink: <https://lore.kernel.org/r/20220610205038.GA3050413@paulmck-ThinkPad-P17-Gen-1>

> BugLink: https://lore.kernel.org/r/CAMdYzYpF4FNTBPZsEFewRuEwSies36QM_As8osPWZSr2q-viEAQ

Other thread discussion about this exact thing:

<https://lore.kernel.org/all/CAHk-=wgzRUT1fBpuz3xcN+YdsX0SxqOzHWrtj0ReHpUBb5TKbA@mail.g>

please stop making up random tags that make no sense.

Just use "Link:"

Look at that first one (I didn't even bother following the second

https://lore.kernel.org/all/CAHk-=wgs38ZrfPvy=nOwVkvzjpM3VFU1zobP37Fwd_h9iAD5JQ@mail.gmail.com/

regtracking with regzbot

so no, we don't use "BugLink:",
"References:" et. al. for this

ideally checkpatch.pl would tell users 'just use "Link:" '

regtracking with regzbot

so no, we don't use "BugLink:",
"References:" et. al. for this

ideally checkpatch.pl would tell users 'just use "Link:" '
anyone volunteering to submit a patch?

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

regzbot's web-ui makes it obvious for me
if things stall before a fix was applied

regtracking with regzbot

regzbot's web-ui makes it obvious for me
if things stall before a fix was applied
and then I'll show up to prod things!

regtracking with regzbot

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

regtracking with regzbot

everybody can interact with regzbot using
commands in a reply to the report

regtracking with regzbot

some of regzbot's commands:

- #regzbot introduced: <commit-id|range>
- #regzbot title: foo
- #regzbot monitor: <https://lore.kernel.org/r/foo@example.com/>
- #regzbot fixed-by: 1f2e3d4c5d
- #regzbot invalid: nothing is broken, by hardware was faulty



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⁾

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

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

next steps for regzbot

- fine tune the recently added support for bugzilla
- making it **self-serving** for subsystem maintainers
- maybe **gitlab support**, to ensure regzbot can monitor DRM bugs, too
- teach regzbot to prod developers automatically if things stall
- quite a few things to make my work easier & lots of optimizations

Some of this will require some bigger internal changes, so it might take a while until they're finished

regtracking with regzbot

regzbot already makes regression tracking a whole lot easier for me already

regtracking with regzbot

it's still tedious work, that's why my regression tracking efforts are far from perfect and on a "best effort basis"

I sometimes ignore issues for one reason or another; I also did quite a few missteps [but learned from them]

regtracking with regzbot

but my efforts afaics
make a real difference

regtracking with regzbot

quite a number of reports would
have fallen through the cracks or
only addressed much later

especially many reports submitted to bugzilla.kernel.org

regtracking with regzbot

thx for sponsoring the regzbot idea:



This website is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871528.

The regzbot project started with funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871528.

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

regtracking with regzbot

thx for sponsoring my current efforts:



[2. best practices for
handling regressions]

best practices

my efforts made me
add two texts on regressions
to the kernel's docs

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:

best practices

the latter outlines
a few best practices for developers
not widely known yet afaics

best practices

(1) please CC: regressions@lists.linux.dev
on replies to regression reports

best practices

(1) please CC: regressions@lists.linux.dev
on replies to regression reports

then everyone including me becomes aware of the report

best practices

(2) when doing so, please consider telling regzbot about the report yourself

The important bits (aka "The TL;DR")

1. Ensure subscribers of the [regression mailing list](mailto:regressions@lists.linux.dev) (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:

```
#regzbot ^introduced 1f2e3d4c5b6a
```

- When forwarding reports from a bug tracker to the regressions list (see above), include a paragraph like the following:

```
#regzbot introduced: v5.13..v5.14-rc1
#regzbot from: Some N. Ice Human <some.human@example.com>
#regzbot monitor: http://some.bugtracker.example.com/ticket?id=1234567
```

3. When submitting fixes for regressions, add "link:" tags to the patch description pointing to

best practices

(3) when fixing regressions, remember to point to the report using a "Link:" tag

Linus wants them, regzbot relies on them

best practices

IOW: when using "Reported-by:",
you in >99% of the cases want to link
to the report using a "Link:" tag, too
ideally, checkpatch.pl would suggest that...

best practices

IOW: when using "Reported-by:",
you in >99% of the cases want to link
to the report using a "Link:" tag, too

ideally, checkpatch.pl would suggest that...
anyone volunteering to submit a patch?

best practices

(4) remember to CC: stable@...,
when needed

(4) remember to CC: stable@...,
when needed

no, a "Fixes:" tag is not enough:
it often will do the trick, but the patch might be
silently dropped if it doesn't apply cleanly!

0 siblings, 1 reply; 3+ messages in thread

From: Greg KH @ 2022-08-24 17:58 UTC ([permalink](#) / [raw](#))

To: Steve French; **+Cc:** Stable, LKML

On Wed, Aug 24, 2022 at 12:14:12PM -0500, Steve French wrote:

> Do changesets that already included the "Fixes:" tag in the commit
> description also need to include the "Cc: stable@vger.kernel.org" in
> order to be included in stable?

As per the documentation:

<https://www.kernel.org/doc/html/latest/process/stable-kernel-rules.html>

You should be putting cc: stable@... on the patch.

But as not all maintainers do, we have to dig through those with Fixes:
in order to actually catch all bugfixes :(

So please, use cc: stable.

thanks,

greg k-h

best practices

(5) fix regressions in a reasonable amount of time!

best practices

what's 'reasonable' obviously
depends on the regression

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>

best practices

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

best practices

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

best practices

short version: often it's a week or two;
for some regressions more time is okay
for others it should just take 2-3 days

best practices

short version: often it's a week or two;
for some regressions more time is okay
for others it should just take 2-3 days
till the fix is merged in mainline!

best practices

to achieve that, keep the
following things in mind

best practices

- (5) fix regressions in reasonable time!
 - (a) Prioritize work wrt. regression over other⁽¹⁾ Linux kernel work

⁽¹⁾ except security vulnerabilities

break `users`".

Because the only thing that matters IS THE USER.

How hard is that to understand?

Anybody who uses "but it was buggy" as an argument is entirely missing the point. As far as the `USER` was concerned, it wasn't buggy - it worked for him/her.

Maybe it worked **because** the `user` had taken the bug into account, maybe it worked because the `user` didn't notice - again, it doesn't matter. It worked for the `user`.

Breaking a `user` workflow for a "bug" is absolutely the WORST reason for breakage you can imagine.

It's basically saying "I took something that worked, and I broke it, but now it's better". Do you not see how f*cking insane that statement is?

And without users, your program is not a program, it's a pointless piece of code that you might as well throw away.

Seriously. This is **why** the #1 rule for kernel development is "we don't break `users`". Because "I fixed a bug" is absolutely NOT AN

best practices

- (5) fix regressions in a reasonable time!
 - (b) regressions in production releases should often have the highest priority and be fixed within a week

best practices

they're thus often more important than regressions only in mainline -rc kernels

developers often can help themselves easily,
users OTOH are often unable to do so

break `users`".

Because the only thing that matters IS THE USER.

How hard is that to understand?

Anybody who uses "but it was buggy" as an argument is entirely missing the point. As far as the `USER` was concerned, it wasn't buggy - it worked for him/her.

Maybe it worked **because** the `user` had taken the bug into account, maybe it worked because the `user` didn't notice - again, it doesn't matter. It worked for the `user`.

Breaking a `user` workflow for a "bug" is absolutely the WORST reason for breakage you can imagine.

It's basically saying "I took something that worked, and I broke it, but now it's better". Do you not see how f*cking insane that statement is?

And without users, your program is not a program, it's a pointless piece of code that you might as well throw away.

Seriously. This is **why** the #1 rule for kernel development is "we don't break `users`". Because "I fixed a bug" is absolutely NOT AN

best practices

- 2022-07-31 Linux 5.19 released
 - **2022-08-11 openSUSE Tumbleweed switches to 5.19**
 - 2022-08-14 Linux 5.19.1 released
 - 2022-08-14 Linux 6.0-rc1 released
 - **2022-08-14 arch Linux switches to 5.19.1**
 - 2022-08-21 5.18.19 & 5.19.3 are released and 5.18.y now EOL
 - **2022-08-28 Fedora switches to 5.19.!!!**
-

best practices

once distros made the jump users have
no easy way to access newly released
versions from the previous stable series

best practices

the previous stable series might also
be close to EOL or EOLed already

best practices

- 2022-07-31 Linux 5.19 released
- 2022-08-11 openSUSE Tumbleweed switches to 5.19
- 2022-08-14 Linux 5.19.1 released
- 2022-08-14 Linux 6.0-rc1 released
- 2022-08-14 arch Linux switches to 5.19.1
- **2022-08-21 5.18.19 & 5.19.3 are released and 5.18.y now EOL**
- 2022-08-28 Fedora switches to 5.19.!!!

best practices

going back to the latest longterm kernel
often is not a option either
as in "not easily available" or "lacks required features" :-/

best practices

sure, users often can stay on their
previous, working kernel; but...

best practices

...remember, we fix security
vulnerabilities every few days

so after a week or two that might be a really bad idea :-/

best practices

that's why regressions found in the latest
productions releases often need to be
fixed faster than those in mainline

best practices

- (5) fix regressions in reasonable time!
 - (c) always consider reverting culprits
-

best practices

it's a often the quickest and easiest way
to resolve a regression

best practices

and prevents more people
running into known issues

best practices

how not to do it

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

best practices

it broke suspend on many machines in
v6.0-rc1..v6.0-rc3

best practices

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

2022-07-19 first bisected regression report[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]

best practices

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

2022-07-19 first bisected regression report[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]

best practices

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

2022-07-19 first bisected regression report[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]

best practices

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>

best practices

- (5) fix regressions in a reasonable time!
 - (d) mainline fixes quickly

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 acceptable to skip testing in linux-next. Especially fixes for regressions in stable and longterm kernels need to be handled quickly, as fixes need to be merged in mainline before they can be backported to older series.

More aspects regarding regressions developers should be aware of

How to deal with changes where a risk of regression is known

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

best practices

I'm regularly seeing fixes lingering on lists or in developer trees for weeks :-/

which is especially bad if the fix needs to be backported to stable...

best practices

maintainers should send changes earlier
or more often upstream⁽¹⁾ when
one of them fixes a regression

⁽¹⁾ e.g. to Linus or their higher level maintainer

best practices

or ask your upstream maintainer⁽¹⁾ to pick up regression fixes directly from the list

⁽¹⁾ Linus won't mind unless you do it every week

best practices

developers should even ask Linus directly
to merge regression fixes sometimes
or example if their subsystem maintainer is MIA

best practices

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

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(-)

best practices

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

best practices

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

best practices

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

best practices

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

best practices

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/>

best practices

BTW, don't let regression fixes wait
till the next merge window⁽¹⁾

⁽¹⁾ unless, of course, there is a strong reason to do so

best practices

I see this regularly, too :-/

which is especially bad if the regression made it into a stable release, so that the fix needs to be backported to it

best practices

such changes in the end afaics get less testing anyway, as many will be backported to stable shortly after rc1

best practices

and even some kernel developers fear testing rc1... :-/

Nathan Chancellor @nathanchance · Aug 15

```
$ uname -sr  
Linux 6.0.0-rc1-llvm
```

Nothing like running a bleeding edge kernel on your workstation :)

2 replies · 38 likes

Christian Brauner @brau_ner

Replying to @nathanchance

Last time I did that was v5.12-rc1 and I was using a swapfile... I had a very fun night... :D

[lwn.net](#)
A warning about 5.12-rc1
Linus Torvalds has sent out a note telling people not to install the recent 5.12-rc1 development kernel; this is especially tr...

6:36 PM · Aug 15, 2022 · Twitter Web App

[3. problems I noticed or face]

topics for discussion;

preface: I have a session on regressions at
the maintainers summit tomorrow

topics for discussion;

I'll try to summarize there what we discuss here – and if you want me to bring something else up there, just let me know

topics for discussion;

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

topics for discussion; on the described state;

on regzbot and it's approach maybe?

topics for discussion; on the described state;

on the described usage of "Link:" tags?

topics for discussion; on the described state;

or the time frames in which regressions should be handled?

I regularly see developers and subsystems that take way longer to address regressions :-/

topics for discussion; on the described state;

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

regzbot, Link tags, expected time frames for fixing, ...

topics for discussion

(2) what to do with bugzilla.kernel.org:
should I watch out for reports there? and
isn't it overdue we do something about it?

[this is something I want to discuss here and then
summarize & discuss tomorrow at the maintainers summit]

topics for discussion; bugzilla;

I noticed quite some good⁽¹⁾
regression reports there that
apparently nobody even 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/>

topics for discussion; bugzilla;

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**
-

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/>

topics for discussion; bugzilla;

note, I don't want to blame anyone for
ignoring bugzilla.kernel.org!

most developers & subsystems never
committed to keep an eye on it!

topics for discussion; bugzilla;

OTOH the current state IMHO is bad for our reputation and scares testers away :-/

many users and even some kernel developers assume wrongly⁽¹⁾ bugzilla.kernel.org is the official place to submit reports, despite `reporting-issues.rst` warning it's not

topics for discussion; bugzilla;

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

topics for discussion; bugzilla;

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

⁽¹⁾ I thus might stop at any time...

topics for discussion; bugzilla;

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...

topics for discussion; bugzilla;

and isn't it long overdue that we fix the
bugzilla.kernel.org mess somehow?

topics for discussion; bugzilla;

I'm pretty sure most users
and quite a few developers would be
happy about any change for the better

topics for discussion; bugzilla;

quick show of hands:

who here thinks that bugzilla.kernel.org
is fine as it is right now?

topics for discussion; bugzilla;

quick show of hands:

who here thinks we should shut it down?

topics for discussion; bugzilla;

quick show of hands:

who here would prefer to keep it
around if someone (who?) at least
improves things somewhat?

topics for discussion; bugzilla;

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

- make someone remove all components and products nobody committed to monitor? and create one that makes "no one will look at this" obvious?
- find minions acting as middleman?
- decommission bugzilla?
- leave things as they are
- [insert suggestion of choice]

topics for discussion;

(3) how do you / how should I /
how should we handle reports from
slightly patched distro kernels?

With that off the table, find below the details on how to properly report issues to the Linux kernel developers.

Make sure you're using the upstream Linux kernel

Are you facing an issue with a Linux kernel a hardware or software vendor provided? Then in almost all cases you are better off to stop reading this document and reporting the issue to your vendor instead, unless you are willing to install the latest Linux version yourself. Be aware the latter will often be needed anyway to hunt down and fix issues.

Like most programmers, Linux kernel developers don't like to spend time dealing with reports for issues that don't even happen with their current code. It's just a waste everybody's time, especially yours. Unfortunately such situations easily happen when it comes to the kernel and often leads to frustration on both sides. That's because almost all Linux-based kernels pre-installed on devices (Computers, Laptops, Smartphones, Routers, ...) and most shipped by Linux distributors are quite distant from the official Linux kernel as distributed by kernel.org: these kernels from these vendors are often ancient from the point of Linux development or heavily modified, often both.

Most of these vendor kernels are quite unsuitable for reporting issues to the Linux kernel developers: an issue you face with one of them might have been fixed by the Linux kernel developers months or years ago already; additionally, the modifications and enhancements by the vendor might be causing the issue you face, even if they look small or totally unrelated. That's why you should report issues with these kernels to the vendor. Its developers should look into the report and, in case it turns out to be an upstream issue, fix it directly upstream or forward the report there. In practice that often does not work out or might not what you want. You thus might want to consider circumventing the vendor by installing the very latest Linux kernel core yourself. If that's an option for you move ahead in this process, as a later step in this guide will explain how to do that once it rules out other potential causes for your issue.

Note, the previous paragraph is starting with the word 'most', as **sometimes developers in fact are willing to handle reports about issues occurring with vendor kernels**. If they do in the end highly depends on the developers and the issue in question. Your chances are quite good if the distributor **applied only small modifications** to a kernel based on a recent Linux version; that for example often holds true for the mainline kernels shipped by Debian GNU/Linux Sid or Fedora Rawhide. Some developers will also accept reports about issues with kernels from distributions shipping the latest stable kernel, as long as its **only slightly modified**; that for example is often the case for **Arch Linux, regular Fedora releases, and openSUSE Tumbleweed**. But keep in mind, **you better want to use a mainline Linux** and avoid using a stable kernel for this process, as outlined in the section 'Install a fresh kernel for testing' in more detail.

Obviously you are free to ignore all this advice and report problems with an old or heavily modified vendor kernel to the upstream Linux developers. But note, those often get rejected or ignored, so consider yourself warned. But it's still better than not reporting the issue at all: sometimes such reports directly or indirectly will help to get the

<https://docs.kernel.org/admin-guide/reporting-issues.html> and
Documentation/admin-guide/reporting-issues.rst

topics for discussion; slightly patches kernels;

esp. bugzilla.kernel.org has many such regressions reports (many of them valid)

topics for discussion; slightly patches kernels;

some developers are happy if I
forward such reports, others
basically told me to ignore them

right now I use my best judgment when deciding what to do

topics for discussion; slightly patches kernels;

(3) how do you /
how should I /
how should we
handle reports from slightly
patched distro kernels?

topics for discussion;

(4) how to ensure reviewers and maintainers prioritize regression fixes?

topics for discussion; slow fixing progress;

I noticed reviewers and subsystems
sometimes fail to notice when
a posted patch fixes a regression

topics for discussion; slow fixing progress;

make it more obvious in the patch description? a free-form tagline like 'Label:' combined with '#regressionfix'?

topics for discussion; slow fixing progress;

[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>

[...]

topics for discussion; slow fixing progress;

tag could be used for other things like
"#nobackport" ("#nostable"?) as well...

topics for discussion; slow fixing progress;

OTOH we could simply try teaching
developers to write better patch
description

topics for discussion; slow fixing progress;

(4) how to ensure reviewers and maintainers prioritize regression fixes?

topics for discussion;

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

topics for discussion; hang...sanitizers;

I currently use my best judgment and ignore some, to keep my load manageable – is that okay (for now)?

topics for discussion;

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

topics for discussion;

(6) how to handle reports from CI systems

topics for discussion; CI reports;

having them in regzbot creates work for me and will likely make the list of tracked issues way longer and harder to read
that's why it might be a bad idea to have them in regzbot

topics for discussion; CI reports;

developer ignore quite some CI reports
because they are not really relevant

another reason why it might be a bad idea
to track them with regzbot

topics for discussion; CI reports;

OTOH it would likely be good to have all regressions listed in one place
but maybe that's something I better should leave for later?

topics for discussion; CI reports;

what do you think?

topics for discussion;

(6) how to handle reports from CI systems

topics for discussion;

(7) take the fear out of reverts somehow?
maybe by allowing culprits of regressions
to be reapplied within reasonable limits?

e.g. even after the merge window

topics for discussion; reverts;

preface: in the end that is
something that needs to be discussed
with Linus tomorrow

topics for discussion; reapply reverts;

I want developers to fix regressions
quickly with a revert
without fearing a big setback

topics for discussion; reverts ;

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

2022-07-19 first bisected regression report[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]

topics for discussion; reverts ;

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>

topics for discussion; reverts;

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

topics for discussion; reverts;

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/>

topics for discussion; reverts ;

both unusually bad, but saw
a few similar situations

in a similar situation I'd submit a revert myself these days

topics for discussion; reverts;

quick reverts could have
saved quite a few people
a lot of time and headaches

topics for discussion; reverts;

but developer apparently often try hard to avoid reverts, as they fear getting that change back in might take time and effort

topics for discussion; reverts;

will bring those two examples up
tomorrow at the maintainers summit and
consider to propose...

topics for discussion; reverts;

...to allow reapplying culprits to mainline
if issue is fixed within ~2 weeks

say only up to -rc5 maybe for any non-crucial change?

topics for discussion; reverts;

or does anyone have a better idea?
or thinks this is stupid?

topics for discussion;

(7) take the fear out of reverts
by allowing culprits of regressions to be
reapplied within reasonable limits?

topics for discussion;

(8) anything else?

finally()

that's it!

Thorsten Leemhuis

Mail: linux@leemhuis.info

GPG Key: 0x72B6E6EF4C583D2D

Twitter: [@kernellogger](#) (en),
[@knurd42](#) (en), [@knurd42rhfc](#) (en),
[@thleemhuis](#) (de) & [@thleemhuisfoss](#) (de)

#EOF