How to report and handle Linux kernel regressions

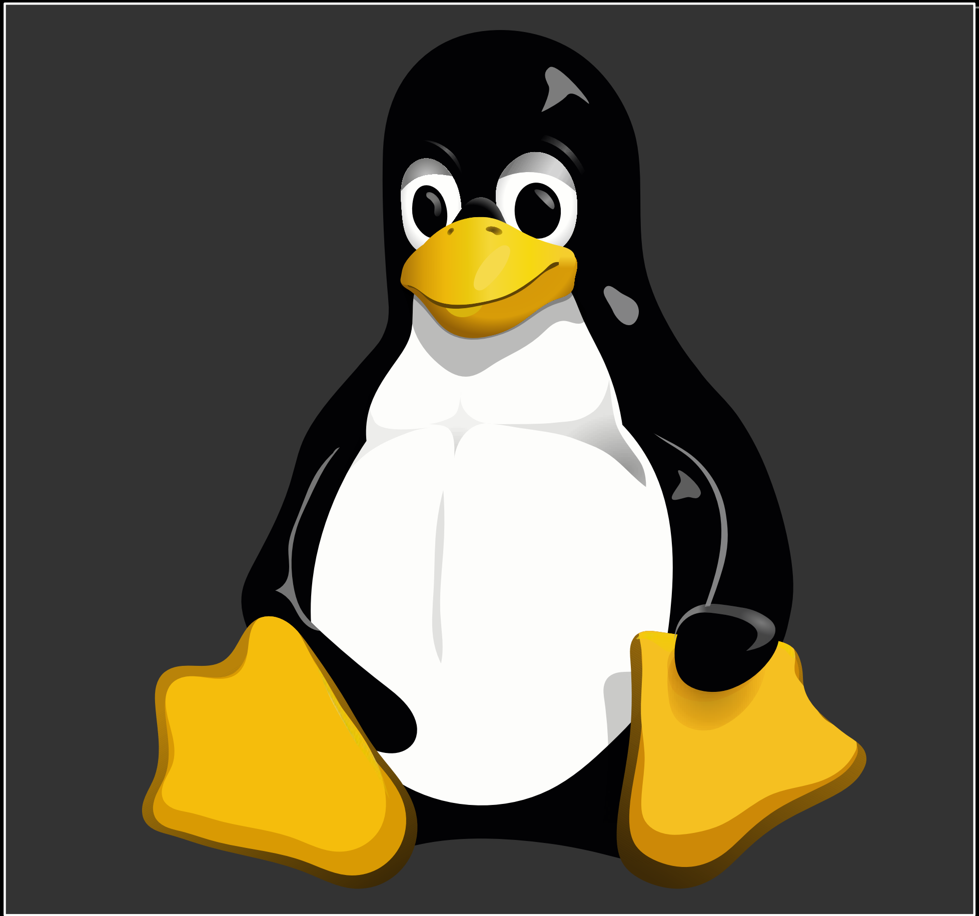
Thorsten Leemhuis

intro;

regressions shall be fixed

NXEU12

intro;



intro;

"no regressions"

intro;

"no regressions"

aka "we don't break user-space"

intro;

even important fixes are reverted,
if it turns out they cause regressions

intro;

looks like quite a few reported regressions are never addressed :-/

intro;

some, because
the report was bad

intro;

some, because
it was a bug and not a regression

intro;

some, because
nobody located the the change causing it

intro;

some, because
developers didn't handle things appropriately

intro;

I'll show you how to
get your regression fixed
as non-developer

intro;

I'll show you how to
handle regressions appropriately
as developer

[1. what exactly is a regression]

definition;

regression == a kernel update breaks something

definition;

It’s a regression if something running fine with one Linux kernel

definition;

It’s a regression if something
running fine with one Linux kernel
works worse or not at all with a newer version

definition;

It’s a regression if something running fine with one Linux kernel works worse or not at all with a newer version that's compiled using a similar configuration

definition;

say your distro updated from v6.1.y to v6.2.y

definition;

say your distro updated from v6.1.y to v6.2.y
and your beloved software from
20 years ago then stops working

definition;

yes, that's a regression

definition;

yes, that's a regression,
as long as it's not caused by an
optional new feature

definition;

yes, that's a regression,
as long as it's not caused by an
optional new feature

definition;

allows progress

definition;

allows introducing new features
like security hardening techniques,
even if they break ancient apps

definition;

you have to explicitly enable such features

definition;

you have to explicitly enable such features at runtime

definition;

you have to explicitly
enable such features
at runtime or build-time

definition;

luckily, new features
known to cause regression are few

definition;

a deliberate config change by your distro
might have broken your beloved app
from 20 years ago

definition;

recheck with a self build kernel

definition;

recheck with self build kernels

[2. someone must locate which change causes the problem]

trailing;

say you pay someone
to upgrade your laptop

trailing;

say you pay someone
to upgrade your laptop and
then something unrelated is broken

trailing;

the person or company you payed has to fix things

trailing;

the developer who caused a regression needs to fix it

the developer who caused
a regression needs to fix it
(or its superior)

trailing;

just report your regression

trailing;

just report your regression –
maybe somebody knows
whats causing it

trailing;

because the root cause is already known

trailing;

because the root cause is already known or because someone can point in the direction of a likely culprit

trailing;

this often works :-D

trailing;

and often it does not :-/

trailing;

and often it does not :-/
– somebody else then
has to locate the culprit

trailing;

in the end you get
what you payed for

trailing;

in the end you get
what you payed for:
nothing

that's why it's your job
as reporter to find the culprit

trailing;

that's why it's your job
as reporter to find the culprit
– which often is needed anyway

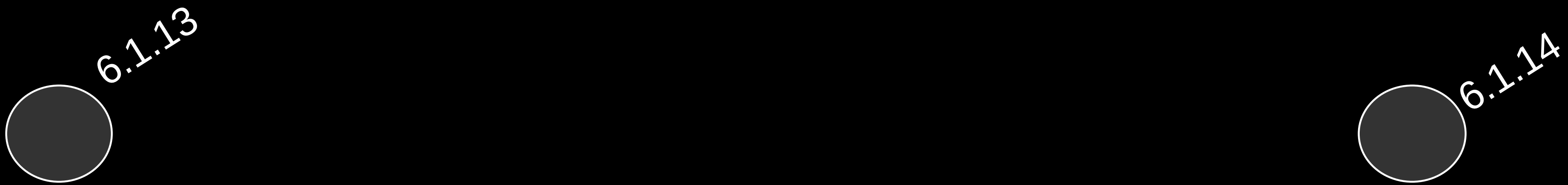
trailing;

culprit can often be found
by compiling ~15 kernels

trailing;

6.1.13

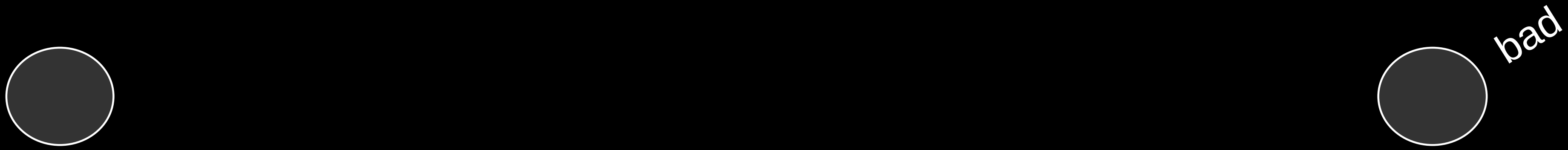
trailing;



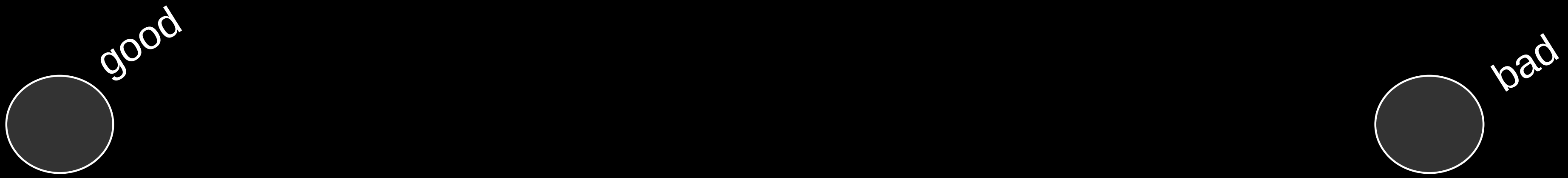
6.1.13

6.1.14

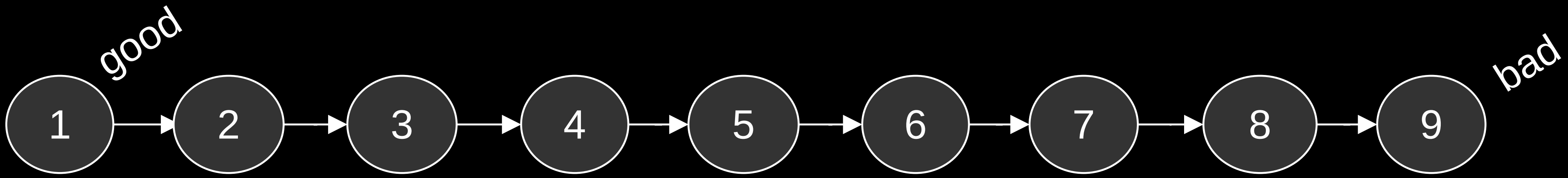
trailing;



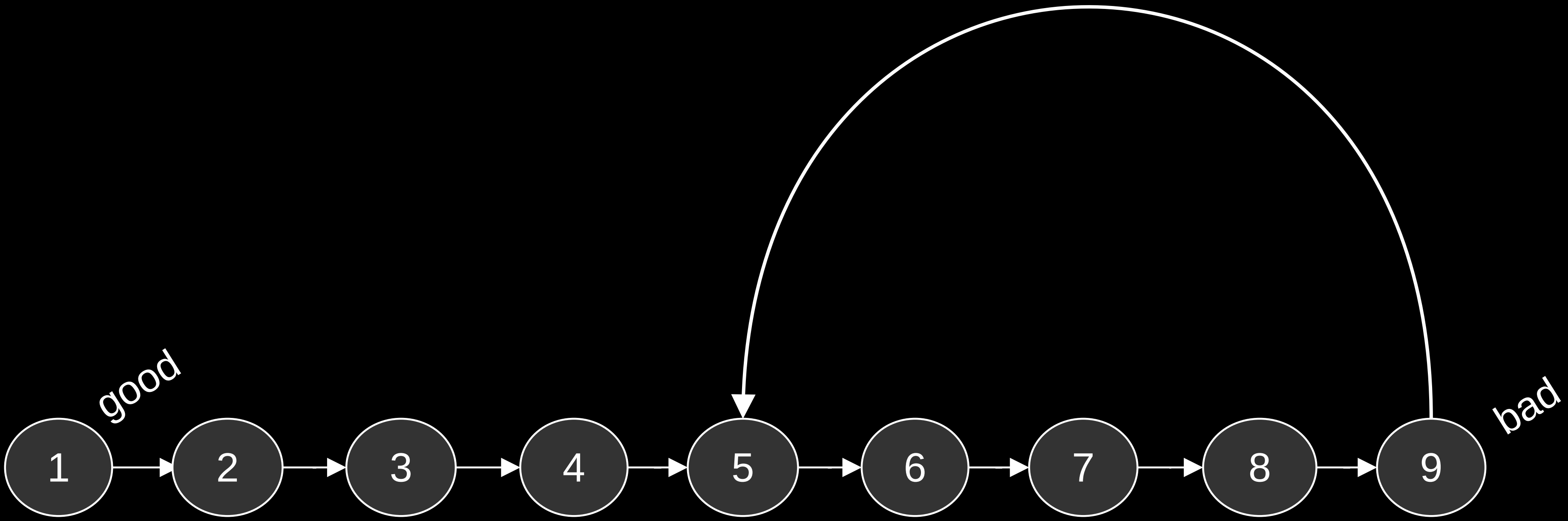
trailing;



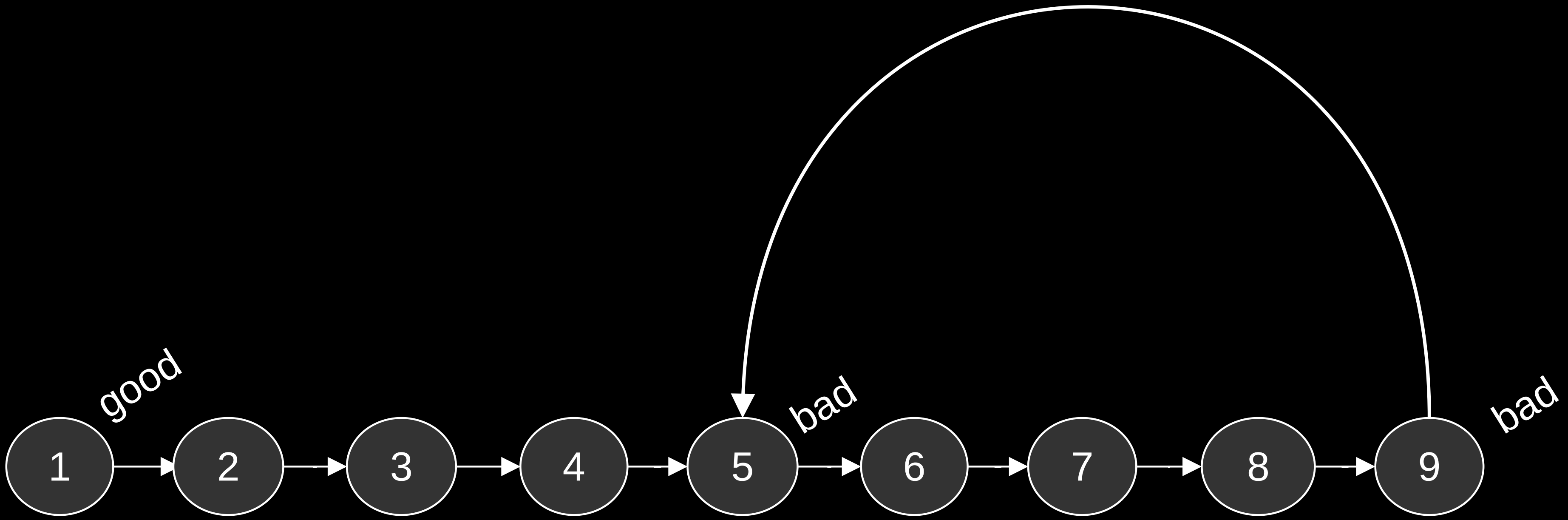
trailing;



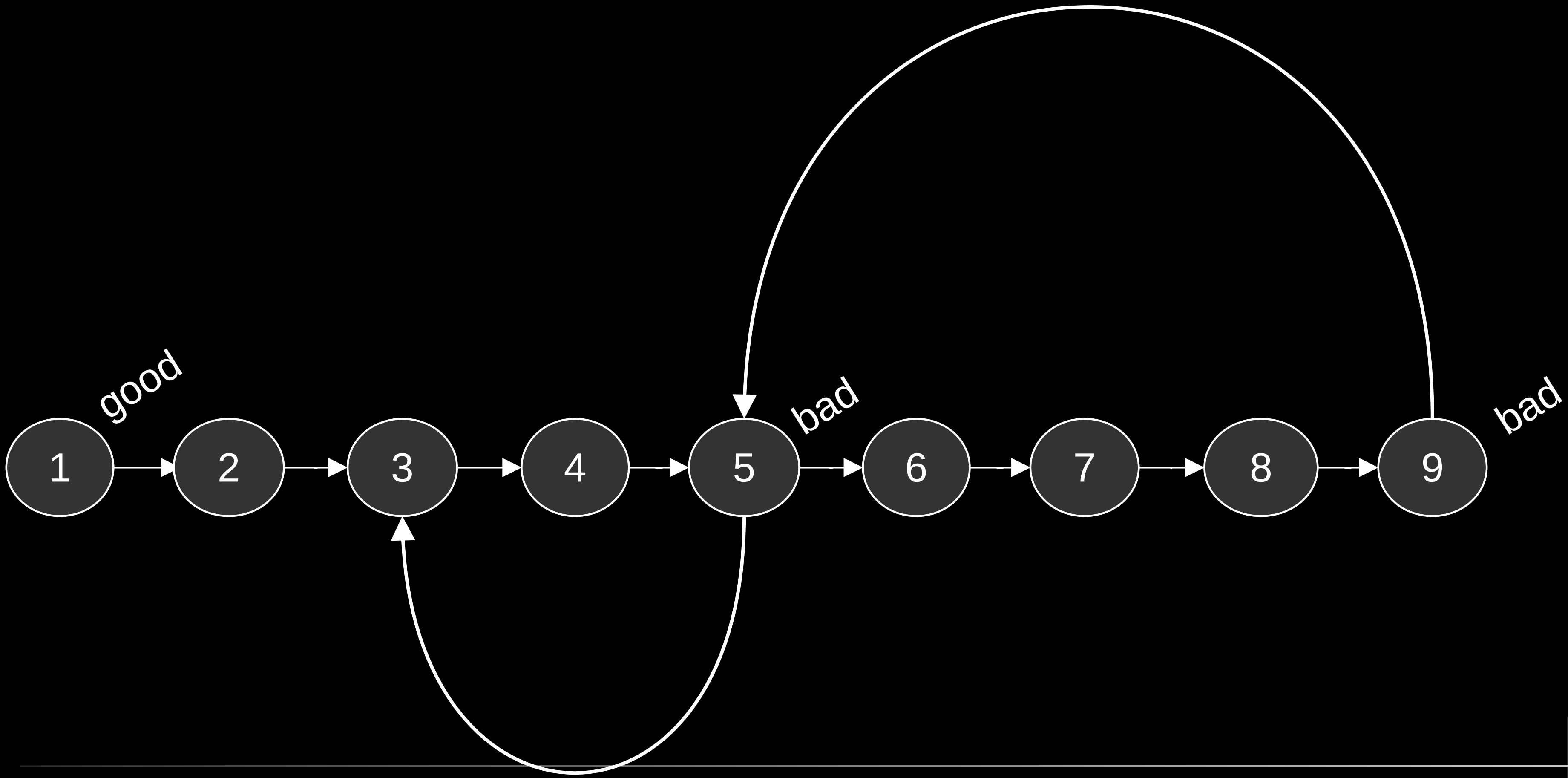
trailing;



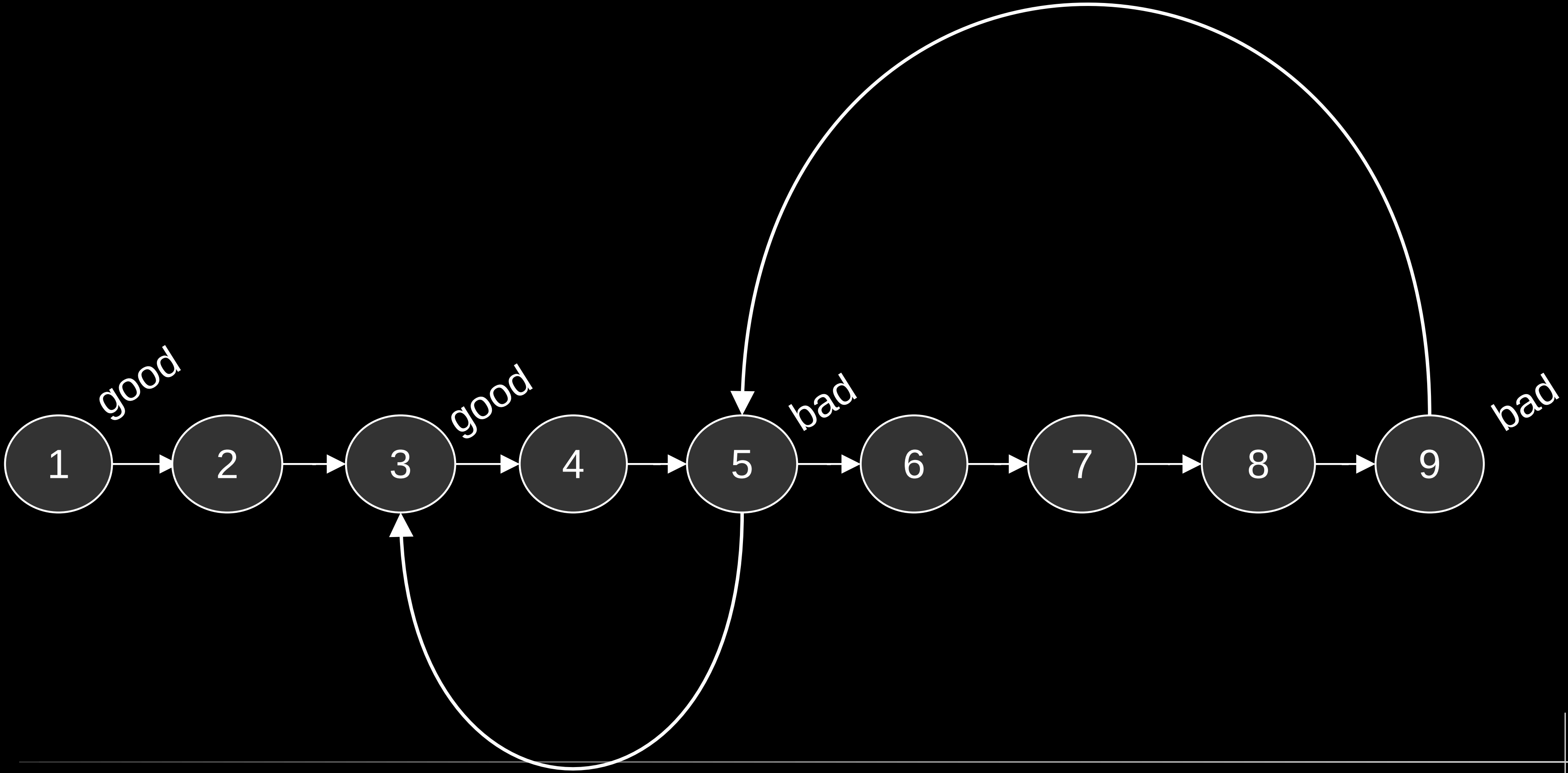
trailing;



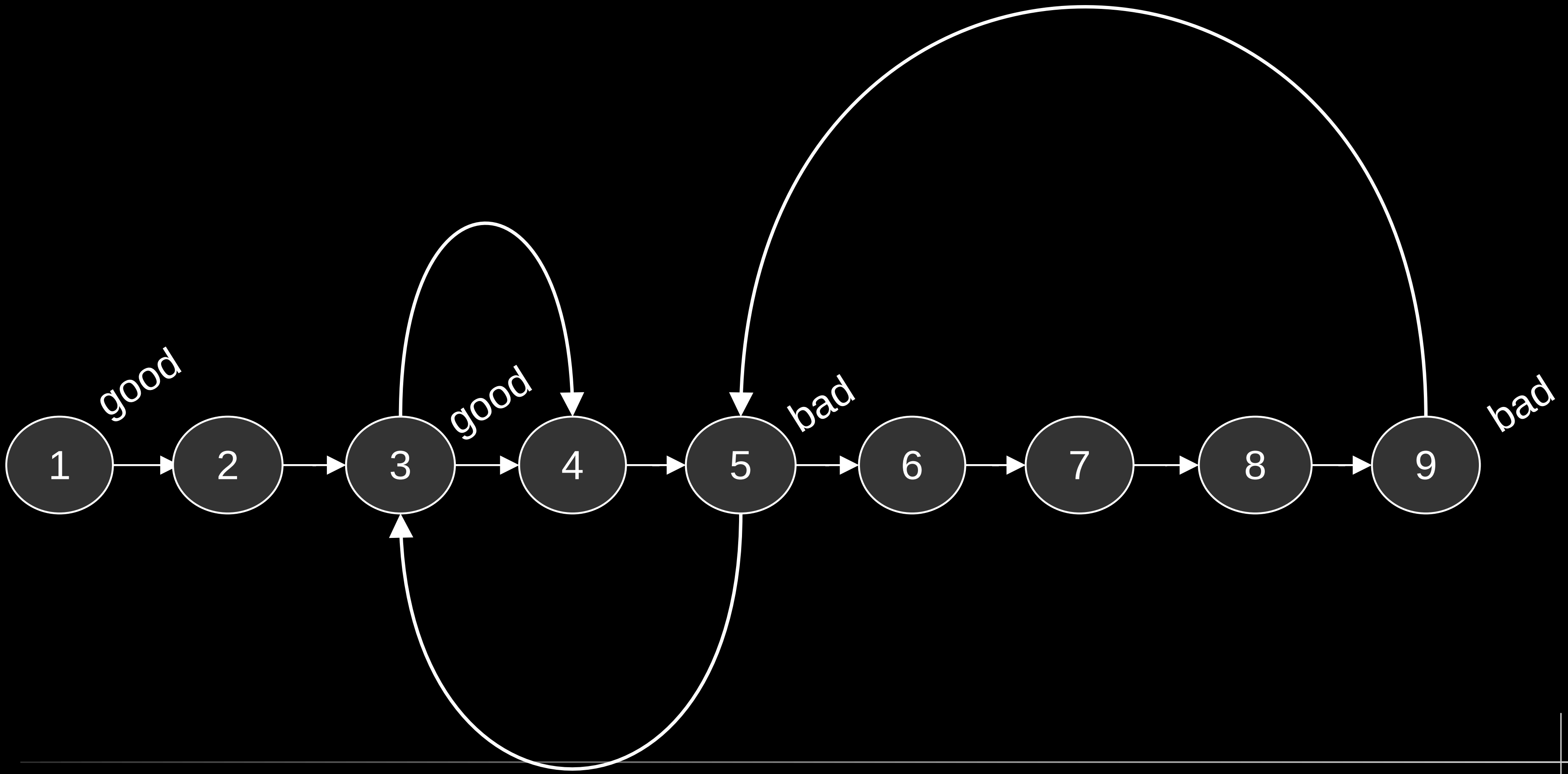
trailing;



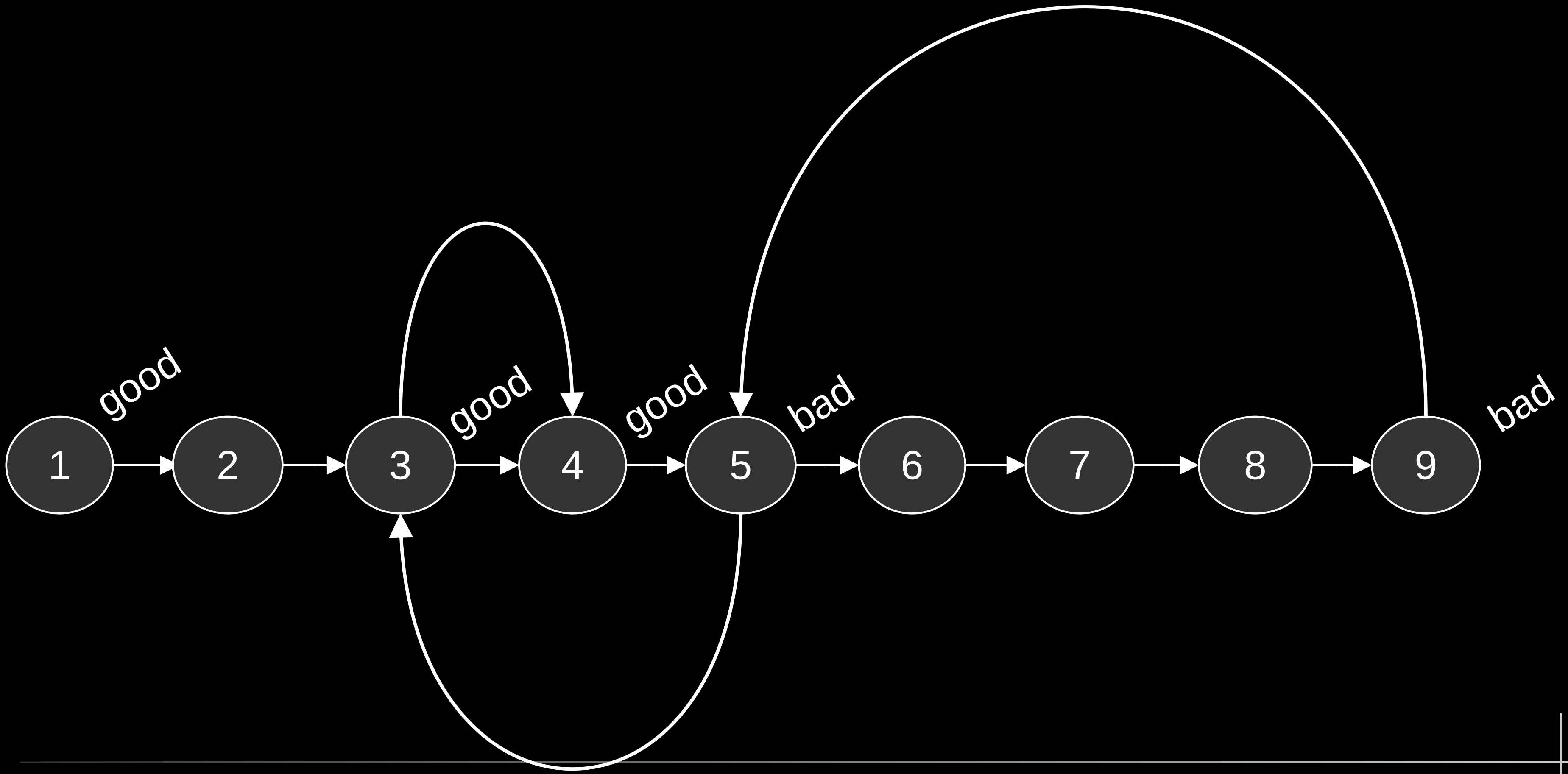
trailing;



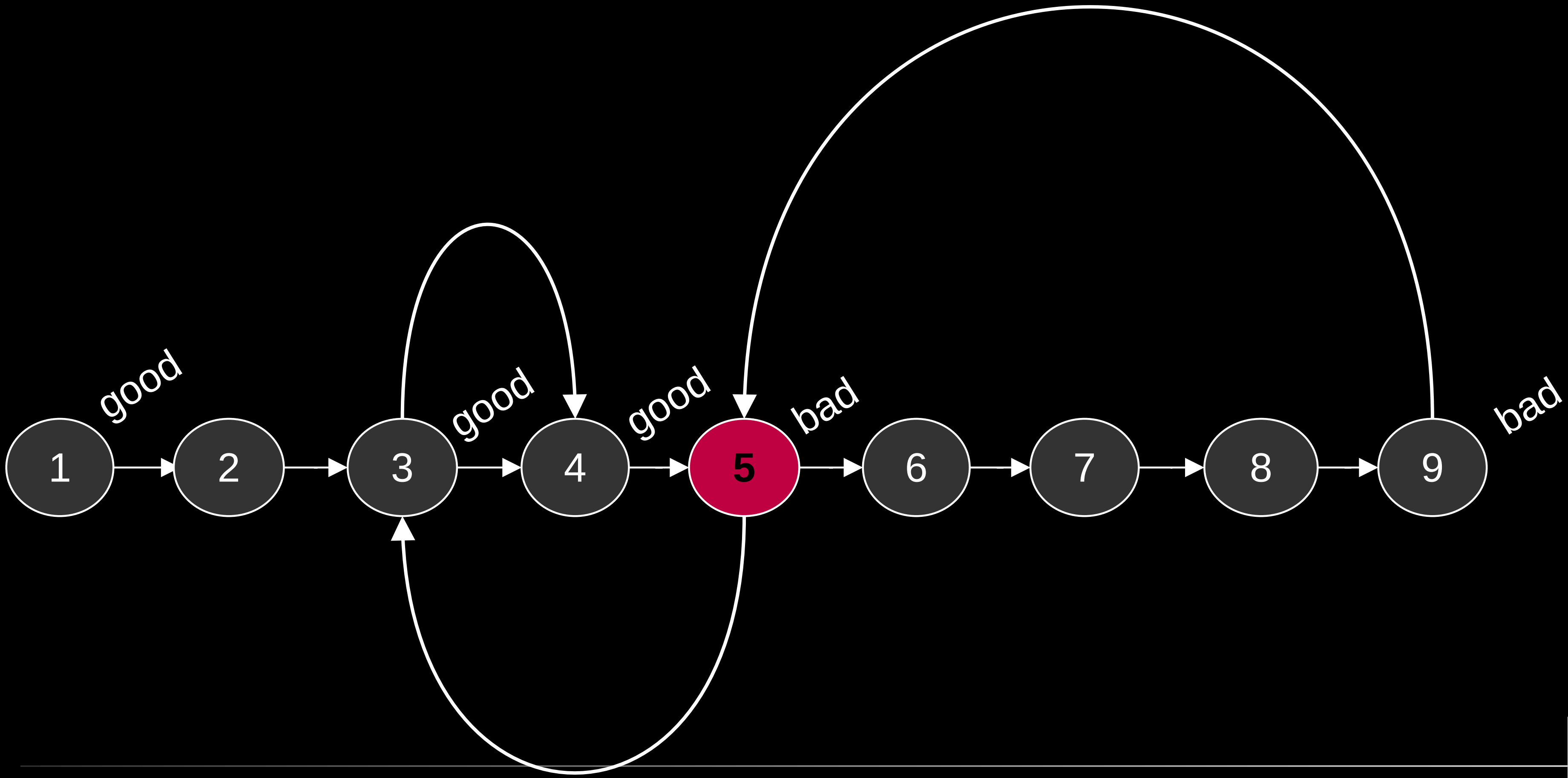
trailing;



trailing;



trailing;



trailing;

once culprit is known,
it's clear who's responsible:

trailing;

once culprit is known,
it's clear who's responsible:
author

trailing;

once culprit is known,
it's clear who's responsible:
author or committer

trailing;

once culprit is known, it's often possible to resolve things quickly

trailing;

once culprit is known, it's often possible to resolve things quickly:

fix

trailing;

once culprit is known, it's often possible to resolve things quickly: fix or revert

trailing;

perform a bisection!

trailing;

perform a bisection!
hands you the lever to get
most regressions quickly fixed

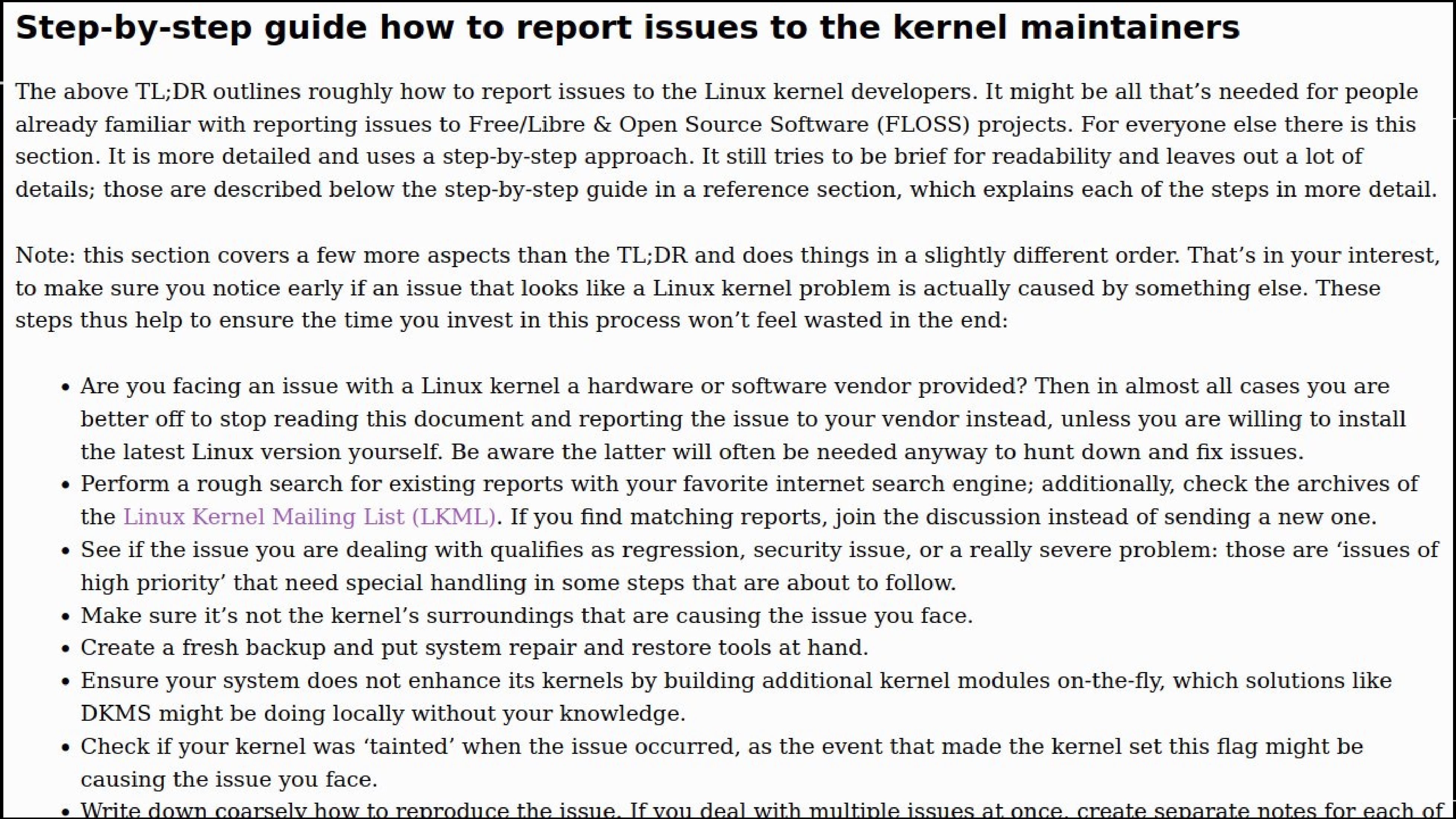
[3. report the problem appropriately]

reporting;

improperly reported regressions might not be fixed

reporting;

Documentation/admin-guide/reporting-issues.rst is your guide



Step-by-step guide how to report issues to the kernel maintainers

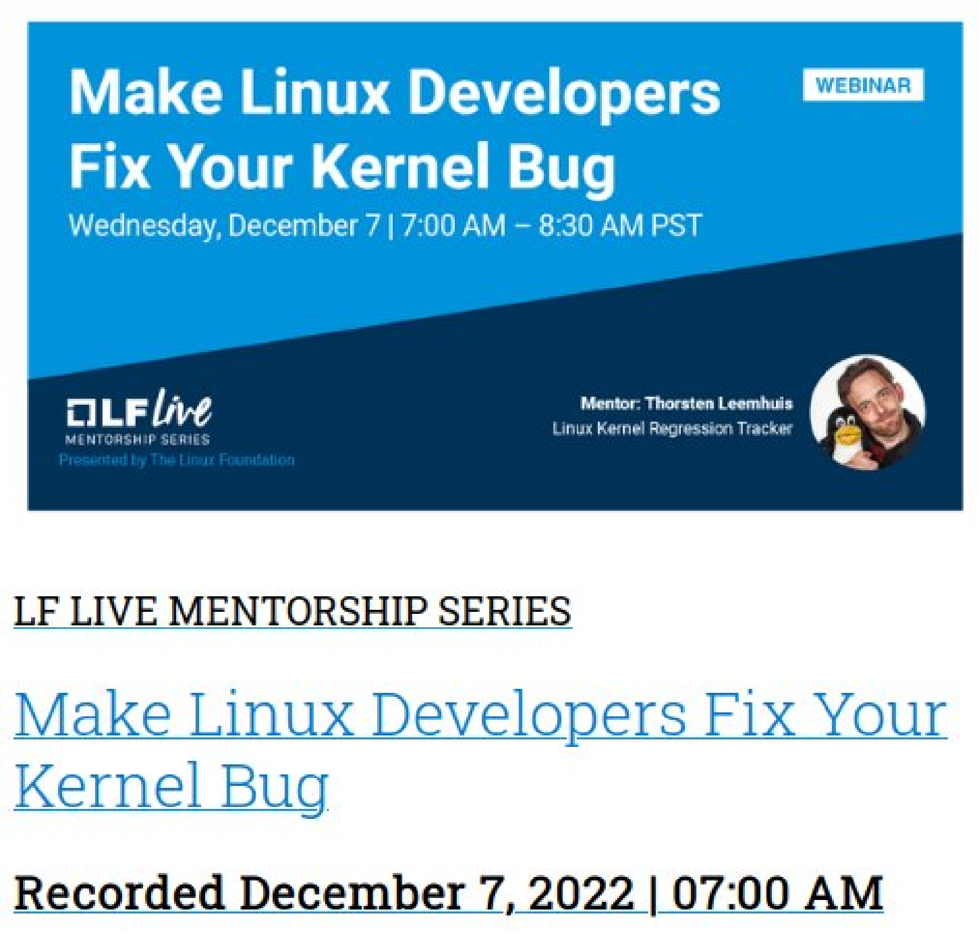
The above TL;DR outlines roughly how to report issues to the Linux kernel developers. It might be all that's needed for people already familiar with reporting issues to Free/Libre & Open Source Software (FLOSS) projects. For everyone else there is this section. It is more detailed and uses a step-by-step approach. It still tries to be brief for readability and leaves out a lot of details; those are described below the step-by-step guide in a reference section, which explains each of the steps in more detail.

Note: this section covers a few more aspects than the TL;DR and does things in a slightly different order. That's in your interest, to make sure you notice early if an issue that looks like a Linux kernel problem is actually caused by something else. These steps thus help to ensure the time you invest in this process won't feel wasted in the end:

- 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.
- Perform a rough search for existing reports with your favorite internet search engine; additionally, check the archives of the Linux Kernel Mailing List (LKML). If you find matching reports, join the discussion instead of sending a new one.
- See if the issue you are dealing with qualifies as regression, security issue, or a really severe problem: those are 'issues of high priority' that need special handling in some steps that are about to follow.
- Make sure it's not the kernel's surroundings that are causing the issue you face.
- Create a fresh backup and put system repair and restore tools at hand.
- Ensure your system does not enhance its kernels by building additional kernel modules on-the-fly, which solutions like DKMS might be doing locally without your knowledge.
- Check if your kernel was 'tainted' when the issue occurred, as the event that made the kernel set this flag might be causing the issue you face.
- Write down coarsely how to reproduce the issue. If you deal with multiple issues at once, create separate notes for each of

https://www.kernel.org/doc/html/latest/admin-guide/reporting-issues.html

reporting;



Make Linux Developers Fix Your Kernel Bug

WEBINAR

Wednesday, December 7 | 7:00 AM – 8:30 AM PST

LF live

MENTORSHIP SERIES

Presented by The Linux Foundation

Mentor: Thorsten Leemhuis

Linux Kernel Regression Tracker

LF LIVE MENTORSHIP SERIES

Make Linux Developers Fix Your Kernel Bug

Recorded December 7, 2022 | 07:00 AM

https://www.linuxfoundation.org/webinars/make-linux-developers-fix-your-kernel-bug

reporting;

1. ensure your kernel is vanilla

reporting;

ensure both working and broken kernels are vanilla

reporting;

as "working" or "broken" might be due to distro modifications

reporting;

2. base your report on a fresh kernel

reporting;

3. ensure your kernel's and system's integrity

reporting;

4. submit your report to the right place

reporting;

5. depict the problem adequately

intro;

To: iwlwifi_maintainer@example.com, iwlwifi_co_maintainer@example.com
CC: iwlwifi_mailinglist@example.com, linux-kernel@vger.kernel.org, regressions@lists.linux.dev
Subject: [Regression] iwlwifi: wifi broken as of Linux 6.0-rc1 ("probe of 0000:00:14.3 failed with error -110")

Hi! Since updating from 5.19.5 to latest mainline (6.0-rc4, vanilla, untainted) my Laptop (Lenovo T14s AMD Gen2 with Fedora 36) my systems doesn't show any WiFi devices anymore. I noticed these error msgs in dmesg:

```
[    2.065312] iwlwifi 0000:00:14.3: enabling device (0000 -> 0002)
[    2.199881] iwlwifi: probe of 0000:00:14.3 failed with error -110
```

Does anyone have an idea what might be wrong here? Or is somebody maybe even working on a fix already? If not I'd be willing to perform a bisection to get down to the root of the problem.

Dmesg: https://example.org/myfiles/dmesg.txt
Kernel-Config: https://example.org/myfiles/config.txt [based on Fedora's]

Ciao, Thorsten

intro;

To: iwlwifi_maintainer@example.com, iwlwifi_co_maintainer@example.com
CC: iwlwifi_mailinglist@example.com, linux-kernel@vger.kernel.org, regressions@lists.linux.dev
Subject: [Regression] iwlwifi: wifi broken as of Linux 6.0-rc1 ("probe of 0000:00:14.3 failed with error -110")

Hi! Since updating from 5.19.5 to latest mainline (6.0-rc4, vanilla, untainted) my Laptop (Lenovo T14s AMD Gen2 with Fedora 36) my systems doesn't show any WiFi devices anymore. I noticed these error msgs in dmesg:

```
[    2.065312] iwlwifi 0000:00:14.3: enabling device (0000 -> 0002)
[    2.199881] iwlwifi: probe of 0000:00:14.3 failed with error -110
```

Does anyone have an idea what might be wrong here? Or is somebody maybe even working on a fix already? If not I'd be willing to perform a bisection to get down to the root of the problem.

Dmesg: https://example.org/myfiles/dmesg.txt
Kernel-Config: https://example.org/myfiles/config.txt [based on Fedora's]

Ciao, Thorsten

intro;

To: iwlwifi_maintainer@example.com, iwlwifi_co_maintainer@example.com
CC: iwlwifi_mailinglist@example.com, linux-kernel@vger.kernel.org, regressions@lists.linux.dev
Subject: [Regression] iwlwifi: wifi broken as of Linux 6.0-rc1 ("probe of 0000:00:14.3 failed with error -110")

Hi! Since updating from 5.19.5 to latest mainline (6.0-rc4, vanilla, untainted) my Laptop (Lenovo T14s AMD Gen2 with Fedora 36) my systems doesn't show any WiFi devices anymore. I noticed these error msgs in dmesg:

```
[    2.065312] iwlwifi 0000:00:14.3: enabling device (0000 -> 0002)
[    2.199881] iwlwifi: probe of 0000:00:14.3 failed with error -110
```

Does anyone have an idea what might be wrong here? Or is somebody maybe even working on a fix already? If not I'd be willing to perform a bisection to get down to the root of the problem.

Dmesg: https://example.org/myfiles/dmesg.txt
Kernel-Config: https://example.org/myfiles/config.txt [based on Fedora's]

Ciao, Thorsten

intro;

To: iwlwifi_maintainer@example.com, iwlwifi_co_maintainer@example.com
CC: iwlwifi_mailinglist@example.com, linux-kernel@vger.kernel.org, regressions@lists.linux.dev
Subject: [Regression] iwlwifi: wifi broken as of Linux 6.0-rc1 ("probe of 0000:00:14.3 failed with error -110")

Hi! Since updating from 5.19.5 to latest mainline (6.0-rc4, vanilla, untainted) my Laptop (Lenovo T14s AMD Gen2 with Fedora 36) my systems doesn't show any WiFi devices anymore. I noticed these error msgs in dmesg:

```
[    2.065312] iwlwifi 0000:00:14.3: enabling device (0000 -> 0002)
[    2.199881] iwlwifi: probe of 0000:00:14.3 failed with error -110
```

Does anyone have an idea what might be wrong here? Or is somebody maybe even working on a fix already? If not I'd be willing to perform a bisection to get down to the root of the problem.

Dmesg: https://example.org/myfiles/dmesg.txt
Kernel-Config: https://example.org/myfiles/config.txt [based on Fedora's]

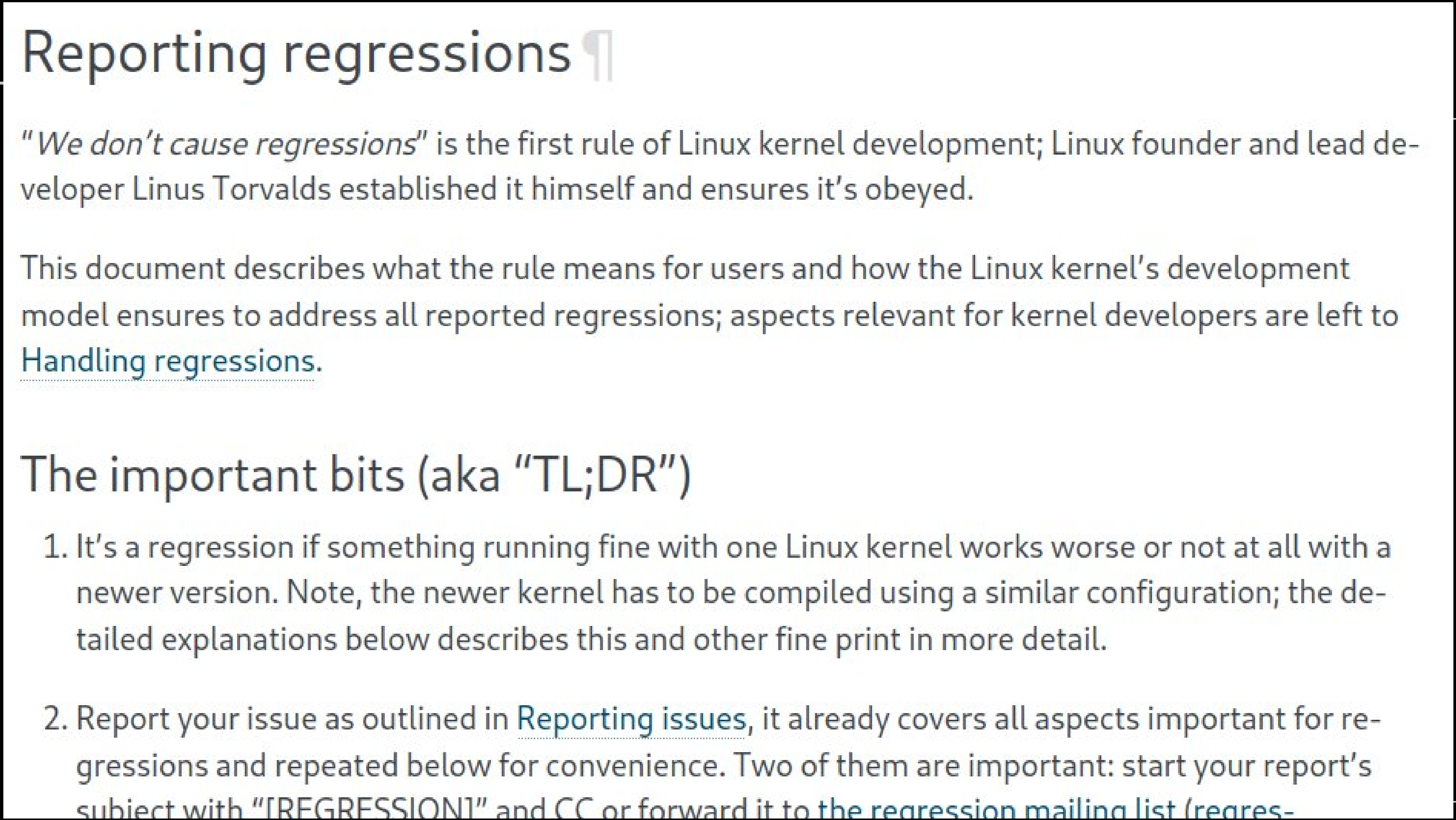
Ciao, Thorsten

reporting;

Documentation/admin-guide/reporting-issues.rst covers everything crucial for regressions

reporting;

Documentation/admin-guide/reporting-regressions.rst holds additional details



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 (regres-

https://docs.kernel.org/admin-guide/reporting-regressions.html

reporting;

1. start your report’s subject with "[REGRESSION]"

reporting;

2. CC or forward it to the regression mailing list

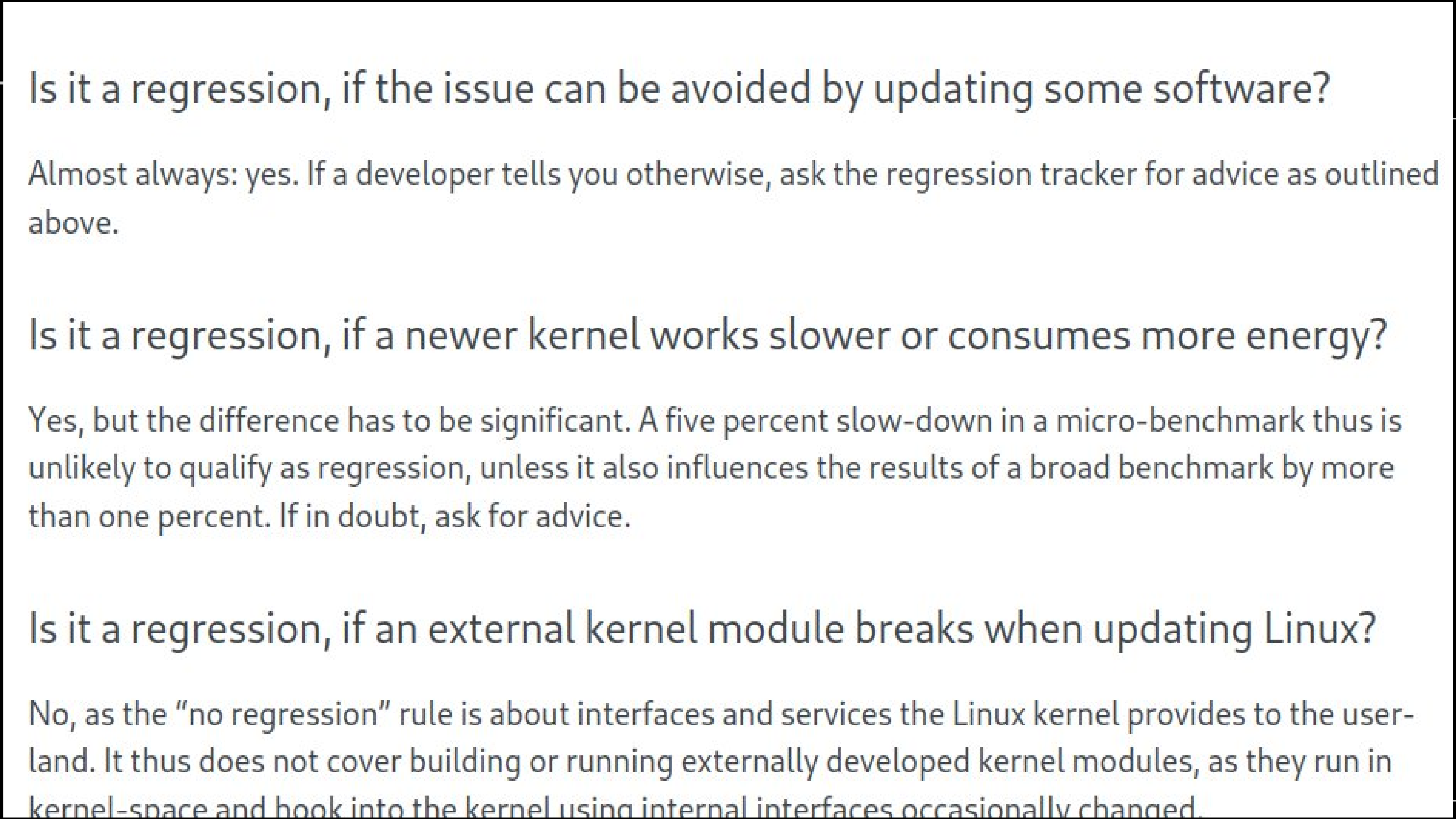
<regressions@lists.linux.dev>

reporting;

[optional]

3. include a paragraph like this:

#regzbot introduced v5.13..v5.14-rc1



Is it a regression, if the issue can be avoided by updating some software?

Almost always: yes. If a developer tells you otherwise, ask the regression tracker for advice as outlined above.

Is it a regression, if a newer kernel works slower or consumes more energy?

Yes, but the difference has to be significant. A five percent slow-down in a micro-benchmark thus is unlikely to qualify as regression, unless it also influences the results of a broad benchmark by more than one percent. If in doubt, ask for advice.

Is it a regression, if an external kernel module breaks when updating Linux?

No, as the "no regression" rule is about interfaces and services the Linux kernel provides to the user-land. It thus does not cover building or running externally developed kernel modules, as they run in kernel-space and hook into the kernel using internal interfaces occasionally changed.

https://docs.kernel.org/admin-guide/reporting-regressions.html

reporting;

Is it a regression, if
a newer kernel works slower or
consumes more energy?

Is it a regression, if
an external kernel module breaks
when updating Linux?

reporting;

Is it a regression, if some test scripts find a API or ABI change?

reporting;

Does the "no regression" rule apply if I seem to be the only person affected?

reporting;

Does the "no regression" rule apply for code in the staging tree as well?

reporting;

What happens if fixing a regression is impossible without causing another?

reporting;

Is it a regression, if
some feature I relied on was
removed months ago?

reporting;

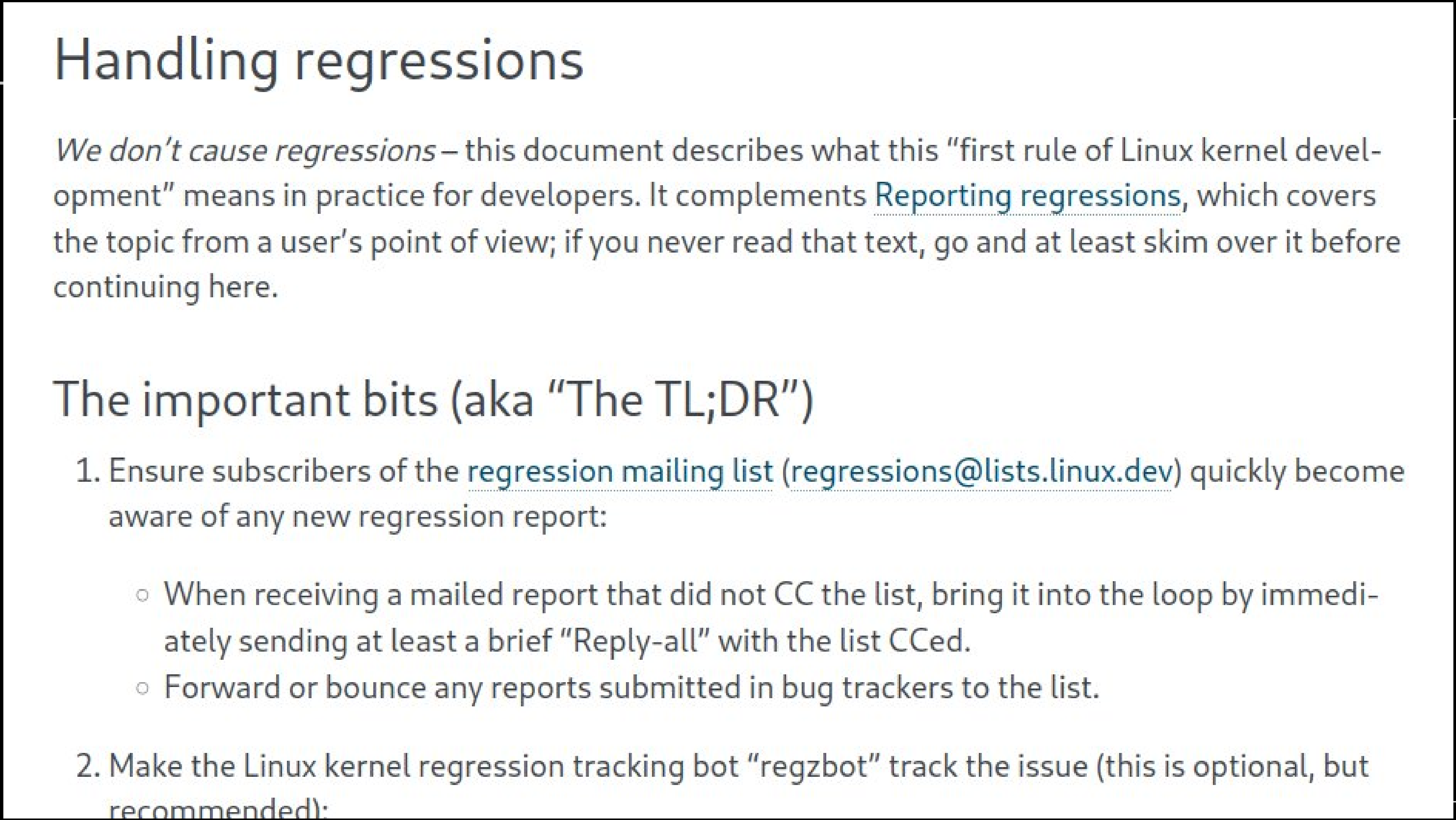
reporting-issues.rst is informative for developers, too

reporting;

reporting-issues.rst is informative for developers, too; but there is a dedicated doc for them as well

reporting;

Documentation/process/handling-regressions.rst



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):

https://docs.kernel.org/process/handling-regressions.html

reporting;

1. CC regressions list

reporting;

[optional]

2. tell regzbot

reporting;

3. when fixing, point to the report using a Link: tag

reporting;

3. when fixing, point to the report using a Link: tag

e.g. like this:

```
Reported-by: Some Human <shuman@example.com>
Link: https://lore.kernel.org/r/123-msgid-456@example.com/
```

reporting;

4. fix regressions quickly

reporting;

fixes for most bisected regressions should be mainlined within 2 weeks

reporting;

many regressions should be mainlined within one week

reporting;

quite a few should be mainlined within two or three days

reporting;

yes, those are ambitious targets

reporting;

yes, those are ambitious targets,
but there are reasons for them

reporting;

see Documentation/process/handling-regressions.rst

reporting;

that document also contains many other insightful things

quotes from Linus on handling regression

reporting;

quotes from Linus
on handling regression
– lots of them

reporting;

also describes how to deal with changes where a regression risk is known

reporting;

or if regzbot should be involved for each and every regression

reporting;

and how to interact with regzbot

[4. regression tracking]

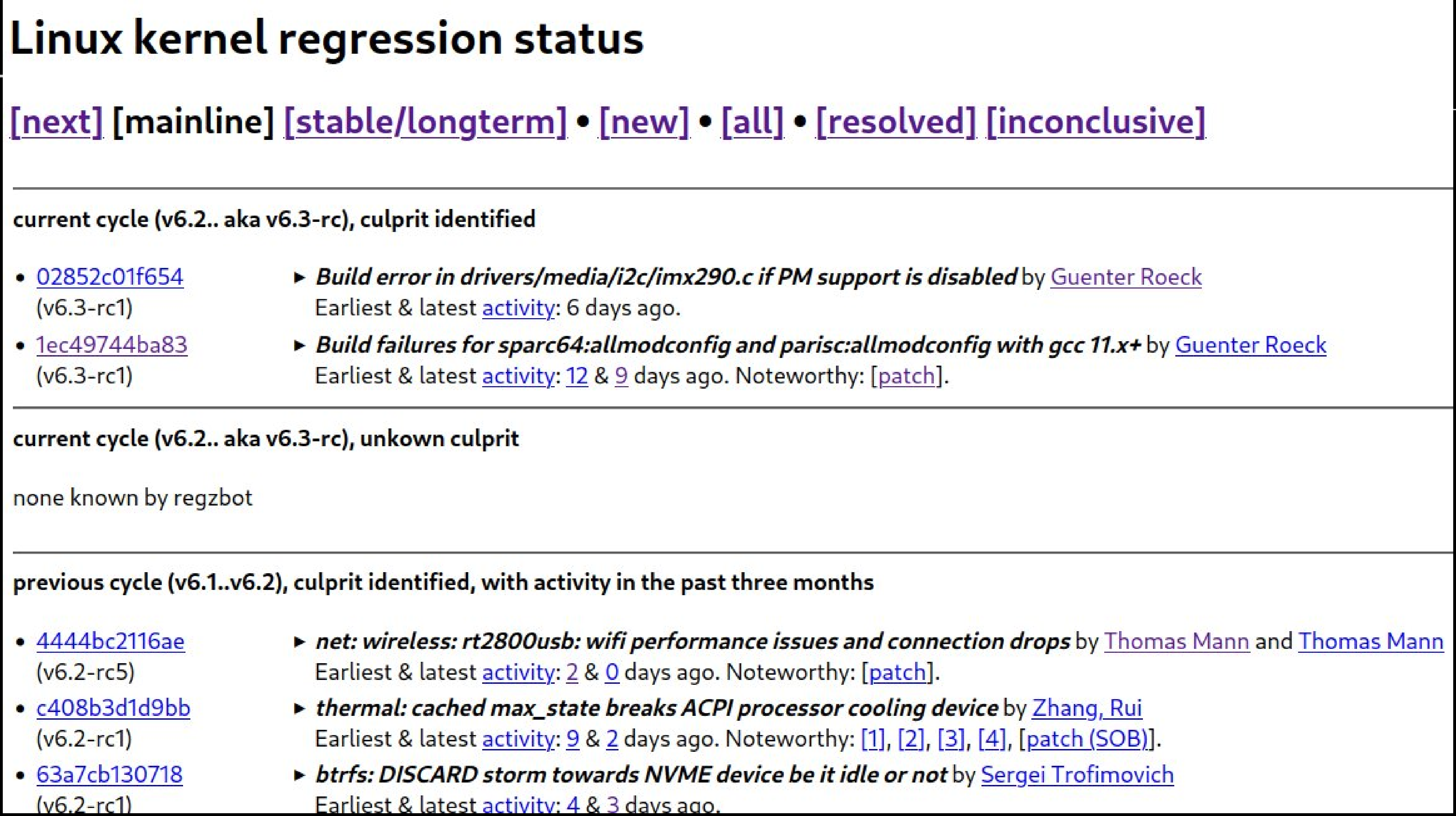
tracking;

helps Linus et. al
doing a better job

```
From: "Regzbot (on behalf of Thorsten Leemhuis)" <regressions@leemhuis.info>
To: LKML <linux-kernel@vger.kernel.org>,
        Linus Torvalds <torvalds@linux-foundation.org>,
        Linux regressions mailing list <regressions@lists.linux.dev>
Subject: Linux regressions report  for mainline [2023-02-05]
Date: Sun,  5 Feb 2023 18:38:38 +0000   [thread overview]
Message-ID: <167562163001.1743361.7332568307990617200@leemhuis.info> (raw)

Hi Linus. Below you'll find regzbot's report about regression from this
cycle me or someone else told the bot about.

Let me highlight three issues you might want to know about:

 * There is a long and ongoing discussion about problems with the NFS
client where there is some disagreement if this qualifies as
regression, as it's a old problem that is more likely to happen now. Not
sure where this heads, a proper fix seems unlikely to emerge quickly. At
least it *currently* looks like this doesn't affect a lot of users.
```

tracking;

ensures the "no regression" rule is no hollow promise



Linux kernel regression status

[next] [mainline] [stable/longterm] • [new] • [all] • [resolved] [inconclusive]

current cycle (v6.2.. aka v6.3-rc), culprit identified

- 02852c01f654 (v6.3-rc1) ▸ ***Build error in drivers/media/i2c/imx290.c if PM support is disabled*** by Guenter Roeck
  Earliest & latest activity: 6 days ago.
- 1ec49744ba83 (v6.3-rc1) ▸ ***Build failures for sparc64:allmodconfig and parisc:allmodconfig with gcc 11.x+*** by Guenter Roeck
  Earliest & latest activity: 12 & 9 days ago. Noteworthy: [patch].

current cycle (v6.2.. aka v6.3-rc), unkown culprit

none known by regzbot

previous cycle (v6.1..v6.2), culprit identified, with activity in the past three months

- 4444bc2116ae (v6.2-rc5) ▸ ***net: wireless: rt2800usb: wifi performance issues and connection drops*** by Thomas Mann and Thomas Mann
  Earliest & latest activity: 2 & 0 days ago. Noteworthy: [patch].
- c408b3d1d9bb (v6.2-rc1) ▸ ***thermal: cached max_state breaks ACPI processor cooling device*** by Zhang, Rui
  Earliest & latest activity: 9 & 2 days ago. Noteworthy: [1], [2], [3], [4], [patch (SOB)].
- 63a7cb130718 (v6.2-rc1) ▸ ***btrfs: DISCARD storm towards NVME device be it idle or not*** by Sergei Trofimovich
  Earliest & latest activity: 4 & 3 days ago.

tracking;

classic bug trackers don't fit well into the Linux kernel's mail based development workflow

tracking;

regzbot in the ideal case just requires *one* additional task

tracking;

when reporting a regression,
add this para to the mailed report:

```
#regzbot introduced v6.2..v6.3-rc1
```

tracking;

when reporting a regression,
add this para to the mailed report:

```
#regzbot introduced 1f2e3d4c5b6b
```

tracking;

regzbot then watches out for replies

tracking;

regzbot then looks out for patches posted to fix tracked regressions

tracking;

regzbot considers regression resolved once fix lands

tracking;

connection made through
Link: tags pointing to the report

tracking;

connection made through
Link: tags pointing to the report

[that's why they are important!]

tracking;

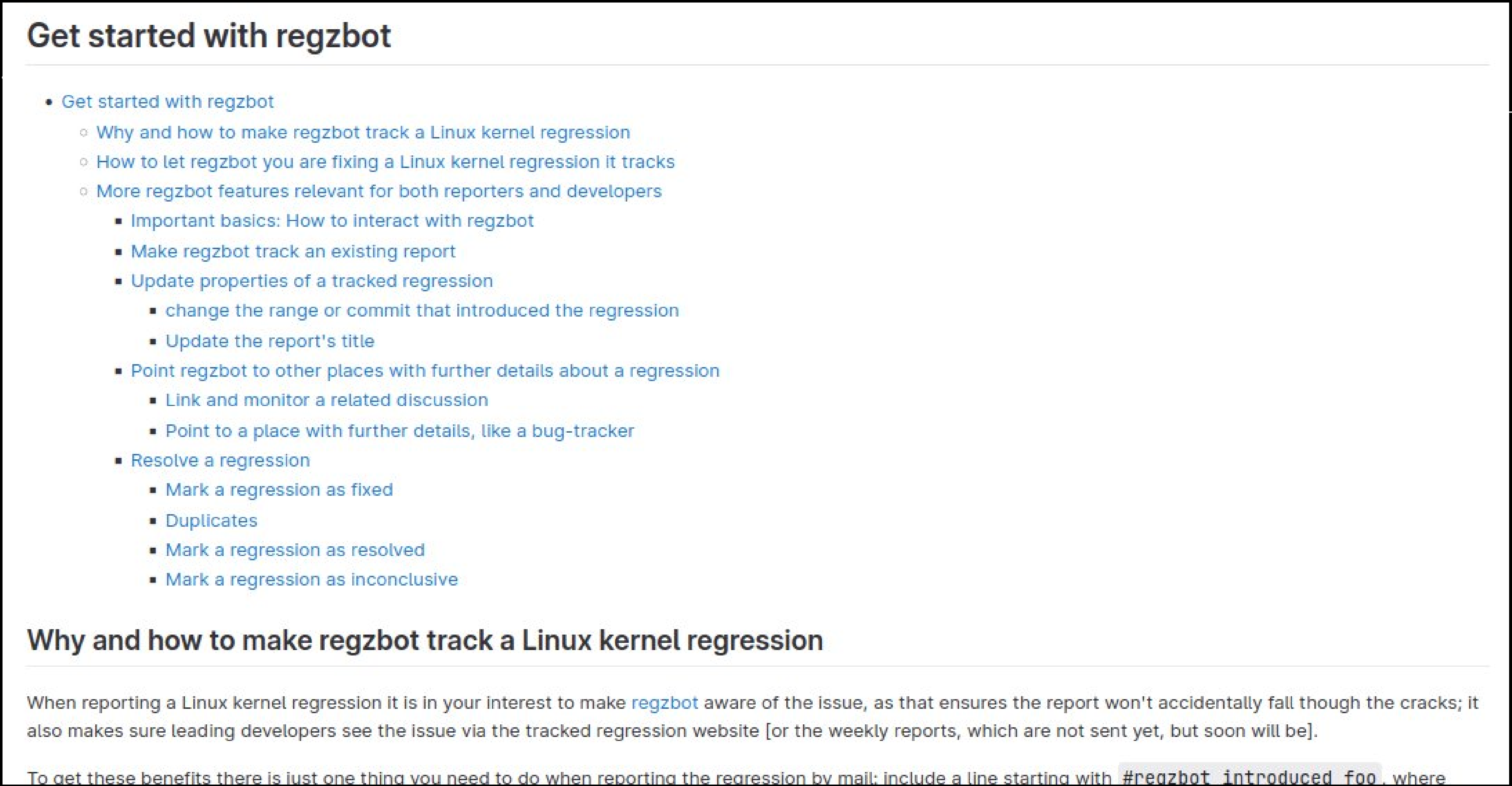
to specify a fix manually, reply to report with a paragraph like this:

```
#regzbot fix 1f2e3d4c5b6a
```

tracking;

to make regzbot track someone else's report, reply with a para like:

```
#regzbot ^introduced v6.2..v6.3-rc1
```



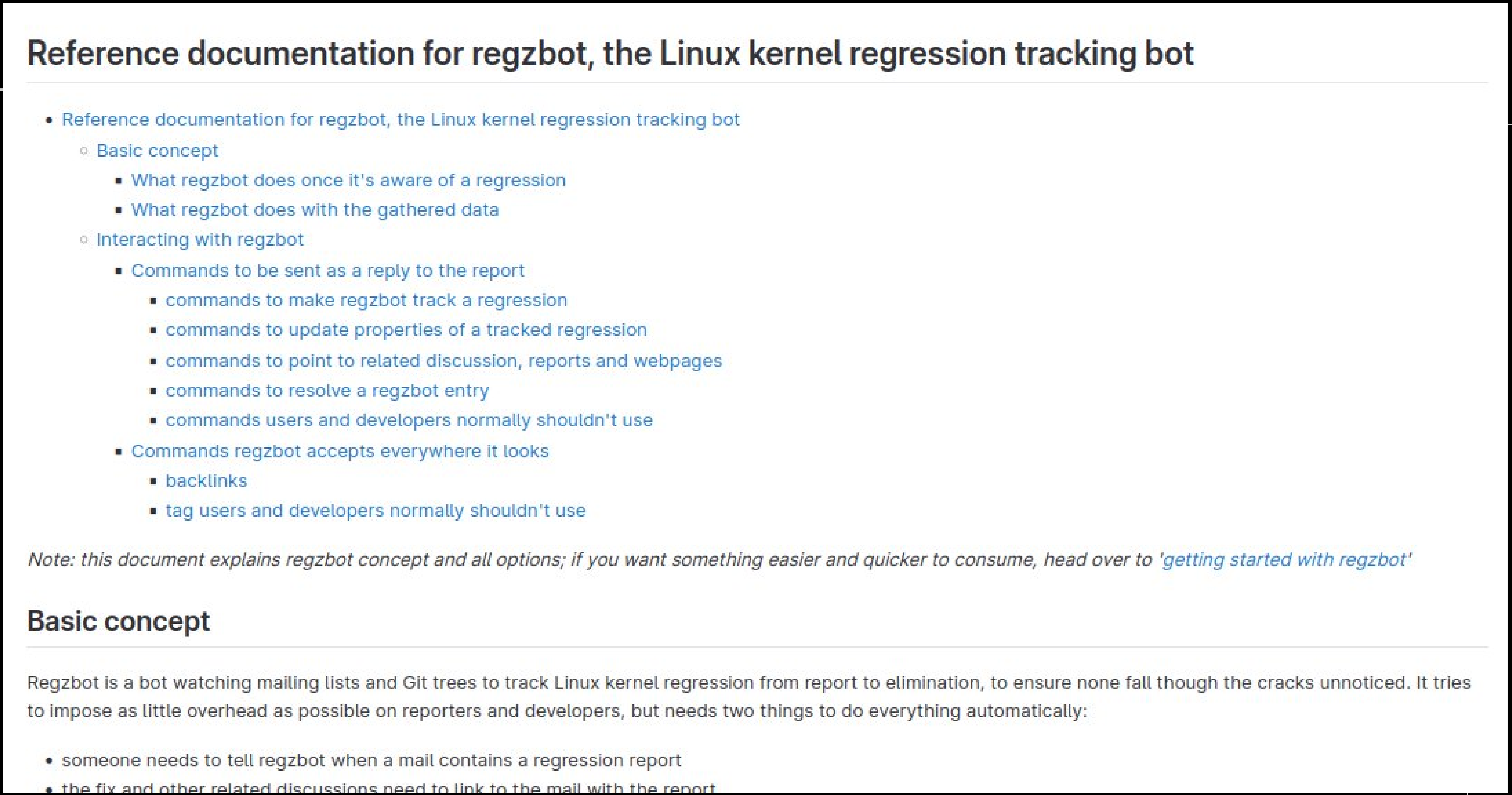
Get started with regzbot

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 though the cracks; it also makes sure leading developers see the issue via the tracked regression website [or the weekly reports, which are not sent yet, but soon will be].

To get these benefits there is just one thing you need to do when reporting the regression by mail: include a line starting with `#regzbot introduced foo`, where

https://gitlab.com/knurd42/regzbot/-/blob/main/docs/getting_started.md



Reference documentation for regzbot, the Linux kernel regression tracking bot

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 though the cracks unnoticed. It tries to impose as little overhead as possible on reporters and developers, but needs two things to do everything automatically:

- someone needs to tell regzbot when a mail contains a regression report
- the fix and other related discussions need to link to the mail with the report

https://gitlab.com/knurd42/regzbot/-/blob/main/docs/reference.md tracking;

thx for sponsoring realizing regzbot:

NGI
POINTER

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/

tracking;

thx for sponsoring my current efforts:

Meta

tracking;

I keep an eye on things with regzbot

tracking;

I keep an eye on things with regzbot
– far from perfect,
but a lot better than nothing!

tracking;

is regression tracking worth it?

tracking;

- Linus seems to like it

tracking;

- Linus seems to like it
- many others said it's great that I do it

;

tracking;

- Linus seems to like it
- many others said it's great that I do it
- helped getting quite few fixes in on the last minute

tracking;

- Linus seems to like it
- many others said it's great that I do it
- helped getting quite few fixes in on the last minute
- regularly brings unfixed regressions back to developers attention

tracking;

I'd say regression tracking is definitely worth it

tracking;

I'd say regression tracking is definitely worth it

[but I'm obviously biased]

tracking;

I'd say regression tracking is definitely worth it

[and there is always something to improve]

[finally()]

finally()

regressions shall be fixed

finally()

my regression tracking tries to ensure that really happens

finally()

hence let the regressions list know about regressions!

finally()

hence let the regressions list know about regressions!

[and ideally regzbot, too]

finally()

contact me if you need
help or advice with anything
wrt to regressions

finally()

developers,
take regression report seriously

finally()

handling-regressions.rst
is your guide

finally()

if you break something,
you need to fix it

finally()

users,
this gives you a pretty long lever

finally()

users,
this gives you a pretty long lever

[but better ensure you use it properly]

finally()

reporting-regressions.rst
is your guide

finally()

bisected the regression

finally()

bisected the regression
with vanilla kernels

finally()

bisected the regression
with vanilla kernels
using a similar configuration

finally()

in initial report,
is totally fine to just offer bisecting!

finally()

remember: there is
no "us versus them" here

finally()

no "users" vs. "developers"

finally()

we are in this together!

Thorsten Leemhuis

mail: linux@leemhuis.info
GPG Key: 0x72B6E6EF4C583D2D

#fediverse: @kernellogger@fosstodon.org (en),
@knurd42@social.linux.pizza (en)

#EOF