

Bisecting Kernel Regressions

Christian Heusel (gromit) - Arch Linux Package Maintainer,
Bug Wrangler & DevOps-Team

Ursprüngliche Folien waren mit jaspervdj/patat präsentiertes
Markdown, alternativer Link (GitHub):

[https://gist.github.com/christian-
heusel/dfafa5007d7d27ea88b4c7d9f4cf8461](https://gist.github.com/christian-heusel/dfafa5007d7d27ea88b4c7d9f4cf8461)

Chapter 0 – Die “No regressions”-Rule

- ▶ <https://docs.kernel.org/admin-guide/reporting-regressions.html>: “It’s a regression if some application or practical use case running fine with one Linux kernel works worse or not at all with a newer version compiled using a similar configuration.”
- ▶ Beachte: Es gibt Kritik bzw. Fußnoten für die Regel
- ▶ Related:
 - ▶ <https://imgs.xkcd.com/comics/workflow.png>
 - ▶ <https://www.hyrumslaw.com/>
- ▶ Linus, wenn regression: (angry-emoji)
<https://lkml.org/lkml/2012/12/23/75>

Chapter 1 – Es gibt einen Bug? :o

- ▶ Fröhliches beta-Testing:

mainline: 6.8-rc1 - 2024-01-21

- ▶ Wahrscheinlicher: Normale Nutzung + Update

*[https://gitlab.archlinux.org/archlinux/packaging/packages/linux/
/issues/](https://gitlab.archlinux.org/archlinux/packaging/packages/linux/issues/)*

Chapter 2 – Na gut, bisecte ich das eben :p

```
$ git rev-list --count v6.7..v6.8-rc1  
13052
```

```
git log rev-list --count v6.8..v6.9  
15678
```

- ▶ git bisect - Use binary search to find the commit that introduced a bug
 - ▶ git bisect start
 - ▶ git bisect bad <ref>
 - ▶ git bisect good <ref>

Chapter 2 – Na gut, bisecte ich das eben :p

- ▶ Auf welcher Commitrange suche ich überhaupt nach dem Bug?
 - ▶ Distribution?
 - ▶ Stable?
 - ▶ Mainline?
- ▶ Was mache ich jetzt also pro revision?

```
$ git bisect start v6.8-rc1
```

```
status: waiting for good commit(s), bad commit known
```

```
$ git bisect good v6.7
```

```
Bisecting: 6785 revisions left to test after this (
```

```
[ba5afb9a84df2e6b26a1b6389b98849cd16ea757] fs: rew
```

Chapter 3 – oh shit, wie teste ich das denn?

PKGBUILD FTW

- ▶ modifiziere `aur/linux-mainline`
 - ▶ baue von `commit` statt `tag`
 - ▶ baue die docs nicht
 - ▶ `pkgver()` einbauen um den `commit` im Paket sichtbar zu machen
- ▶ integration mit `mkinitcpio` etc.
- ▶ offload-build (hehe)
- ▶ Resultat:
<https://pkgbuild.com/~gromit/linux-bisection-kernels/>

Chapter 3 – oh shit, wie teste ich das denn?

mkosi

- ▶ kann grob das folgende:
 1. Baue eine OS Dateistruktur auf
 2. Package das in ein output format
 3. (optional) Boote via `systemd-nspawn` oder `qemu`
- ▶ Unterstützt gängige Linux Distributionen (unabhängig des “Hosts”)

Chapter 3 – oh shit, wie teste ich das denn?

mkosi

mkosi-kernel

mkosi.local.conf:

[Content]

BuildSources=/home/chris/Documents/shared_projects/linux

[Distribution]

Distribution=debian

Chapter 4 – Bugreport

Commit: 7a36b901a6eb0e9945341db71ed3c45c7721cfa9

("ACPI: OSL: Use a threaded interrupt handler for SCI")

- ▶ bug reporting guidelines lesen
 - ▶ ...
 - ▶ Reproduzierbarkeit sicherstellen
 - ▶ Distropatches ausschließen
 - ▶ Nach Duplikaten suchen
 - ▶ Neuste Version checken
 - ▶ Möglichst viel Infos dazupacken
 - ▶ MAINTAINERS file für bug report checken
 - ▶ Optional: Regzbot (<https://linux-regtracking.leemhuis.info/regzbot/mainline/>)

Ergebnis: https://bugzilla.kernel.org/show_bug.cgi?id=218407

Chapter 4 – Bugreport

- ▶ es passiert ... nichts?
- ▶ Poste bugreport unter den Patch, aber wie?
<https://lore.kernel.org/all/5745568.DvuYhMxLoT@kreaacher/#t>
- ▶ Bugreport an die Regressions Liste schicken und maintainer CC'en (`get_maintainer.pl`)
- ▶ Bugreport ist nicht fertig, für Rückfragen & Debugpatches erreichbar bleiben

Chapter 5 – Lösung

```
--- a/drivers/pinctrl/pinctrl-amd.c
```

```
+++ b/drivers/pinctrl/pinctrl-amd.c
```

```
@@ -1162,7 +1162,7 @@ static int amd_gpio_probe(struct  
    }
```

```
        ret = devm_request_irq(&pdev->dev, gpio_dev->irq,
```

```
-                                IRQF_SHARED, KBUILD_MODNAME,
```

```
+                                IRQF_SHARED | IRQF_ONESHOT,
```

```
        if (ret)
```

```
            goto out2;
```

```
--
```

2.34.1

Test:

<https://lore.kernel.org/all/q336uhrwuvhaf2x4fc6tneaavgugcy-szgn75vzbrr4ksf7oxhi@3qcwff6nuvei/>

Zusätzliche Ressourcen

- ▶ #kernelnewbies on oftc.net IRC network
- ▶ <https://www.kernel.org/doc/html/latest/admin-guide/reporting-issues.html>
- ▶ <https://docs.kernel.org/admin-guide/verify-bugs-and-bisect-regressions.html>
- ▶ Daan De Mayer (FOSDEM2024): **Streamlining kernel hacking with mkosi-kernel**
<https://fosdem.org/2024/schedule/event/fosdem-2024-2209-streamlining-kernel-hacking-with-mkosi-kernel/>
- ▶ **A re-introduction to mkosi – A Tool for Generating OS Images** <https://0pointer.net/blog/a-re-introduction-to-mkosi-a-tool-for-generating-os-images.html>