# LF live
MENTORSHIP SERIES

LF live MENTORSHIP SERIES

# 2nd Place is the 1st Winner

## Growing Your Influence in the Linux Kernel Community

Dan Williams, Principal Engineer, Intel

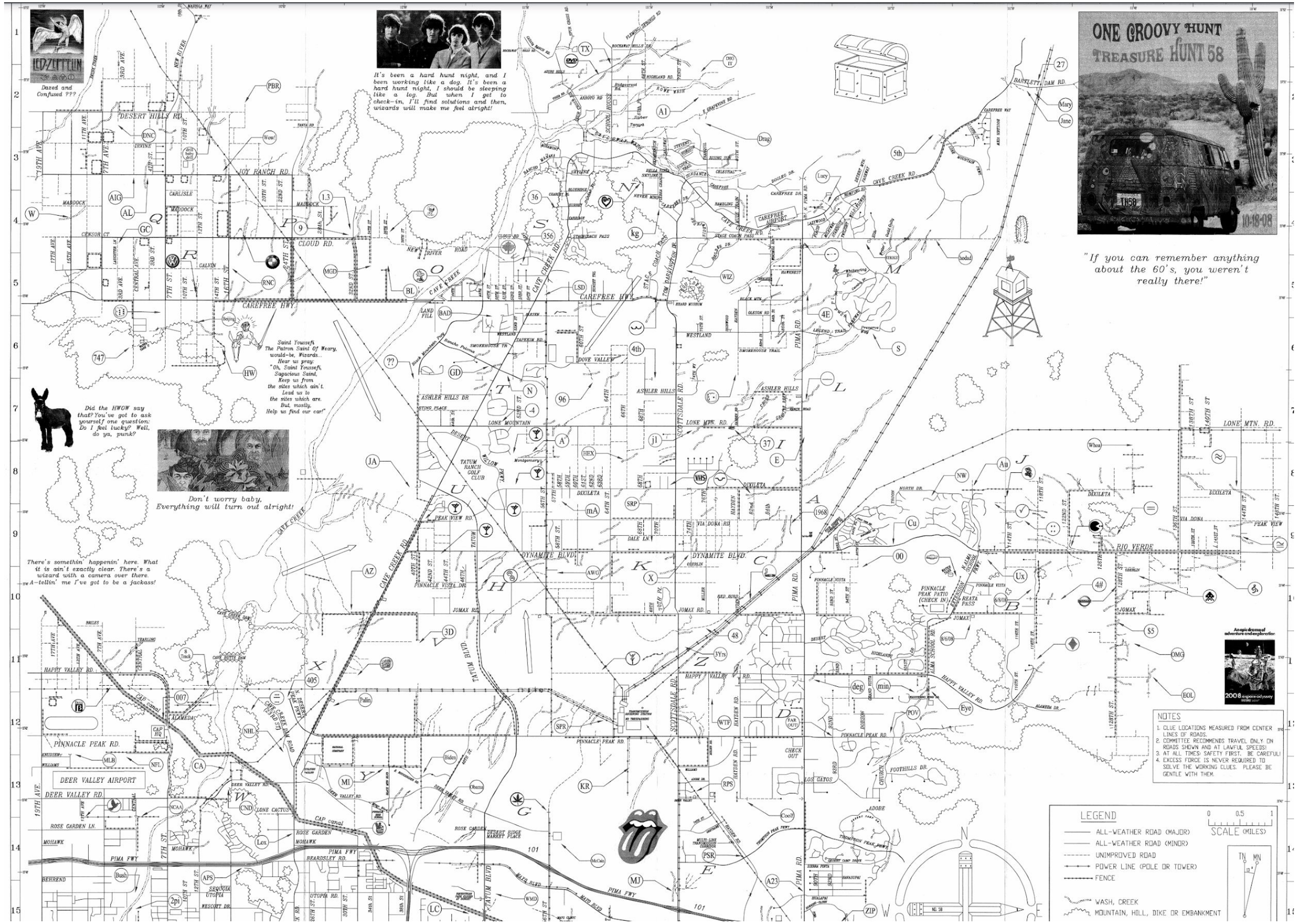
## Agenda

- Career Path Disclaimer
- 2nd Place is the 1st Winner
- How to Kernel
  - Review
  - Maintain
- Conclusion / why this is worth the effort

THE LINUX FOUNDATION

## 2nd Place is the 1st Winner

ONE GROOVY HUNT
TREASURE HUNT 58
ARIZONA
TH58
10-18-08



Dazed and Confused ???

*It's been a hard hunt night, and I been working like a dog. It's been a hard hunt night, I should be sleeping like a log. But when I get to check-in, I'll find solutions and then, wizards will make me feel alright!*

"If you can remember anything about the 60's, you weren't really there!"

*Saint Youssefi*
*The Patron Saint Of Weary, would-be, Wizards...*
*Hear us pray:*
*"Oh, Saint Youssefi, Sagacious Saint, Keep us from the sites which ain't. Lead us to the sites which are. But, mostly, Help us find our car!"*

*Did the HWOW say that? You've got to ask yourself one question: Do I feel lucky? Well, do ya, punk?*

*Don't worry baby, Everything will turn out alright!*

*There's somethin' happenin' here. What it is ain't exactly clear. There's a wizard with a camera over there. A-tellin' me I've got to be a jackass!*

NOTES

1. CLUE LOCATIONS MEASURED FROM CENTER LINES OF ROADS.
2. COMMITTEE RECOMMENDS TRAVEL ONLY ON ROADS SHOWN AND AT LAWFUL SPEEDS!
3. AT ALL TIMES: SAFETY FIRST. BE CAREFUL!
4. EXCESS FORCE IS NEVER REQUIRED TO SOLVE THE WORKING CLUES. PLEASE BE GENTLE WITH THEM.

LEGEND

| | |
|---|---|
| —— | ALL-WEATHER ROAD (MAJOR) |
| —— | ALL-WEATHER ROAD (MINOR) |
| - - - - | UNIMPROVED ROAD |
| •—•—• | POWER LINE (POLE OR TOWER) |
| ×—×—× | FENCE |
| 〰 | WASH, CREEK |
| ⩘⩘ | MOUNTAIN, HILL, DIKE OR EMBANKMENT |

0 0.5 1
SCALE (MILES)

TN MN 13°

## 2nd Place is the 1st Winner

- If you race to maintainership you might unfortunately win
- Kernel maintainer is a support role, not an executive role
- The power to say “no” comes with the responsibility to say “no”

LF Live MENTORSHIP SERIES

## How to Kernel: Review

## How to Kernel: Getting Started

- Mentor Programs
- Fix a bug or 10
- Scratch an itch

## How to Kernel: Getting Started

- Mentor Programs
- Fix a bug or 10
- Scratch an itch

- Add a new feature

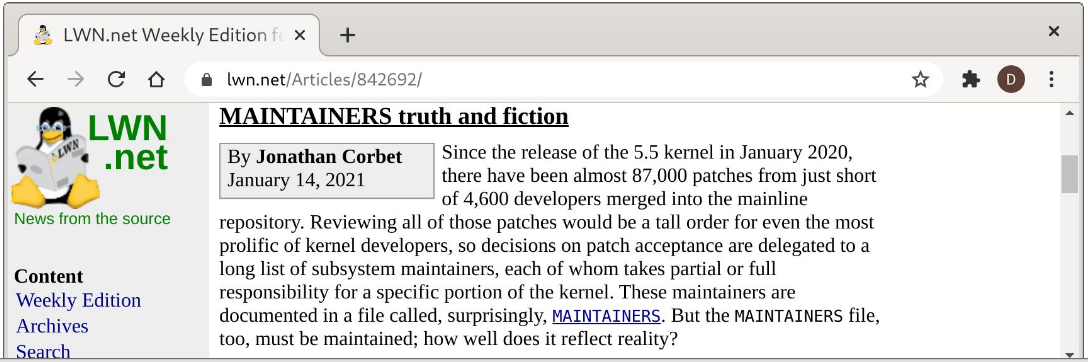
LF Live MENTORSHIP SERIES

## How to Kernel: Getting Started

- Mentor Programs
- Fix a bug or 10
- Scratch an itch

- Add a new feature

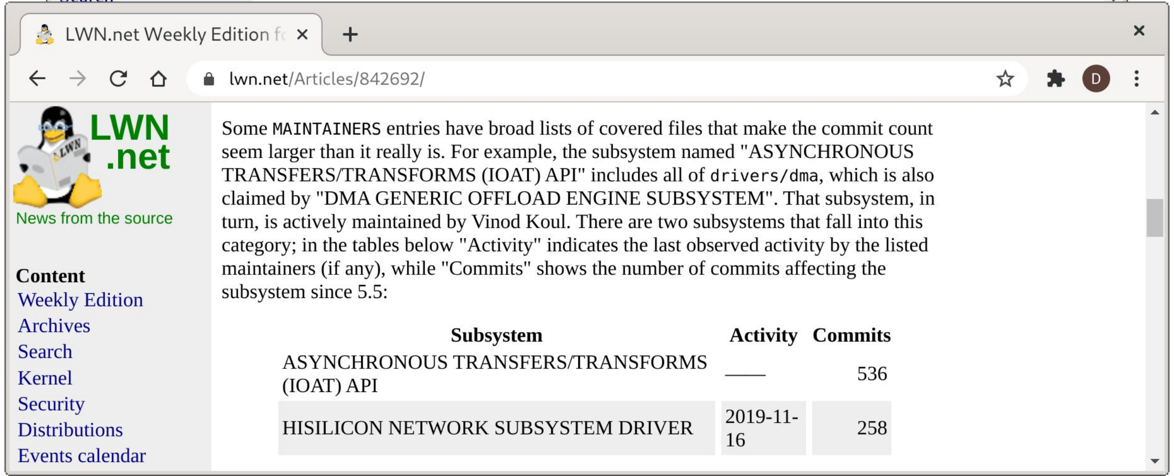
Graduate to a Reviewer not Maintainer



## MAINTAINERS truth and fiction

By **Jonathan Corbet**
January 14, 2021

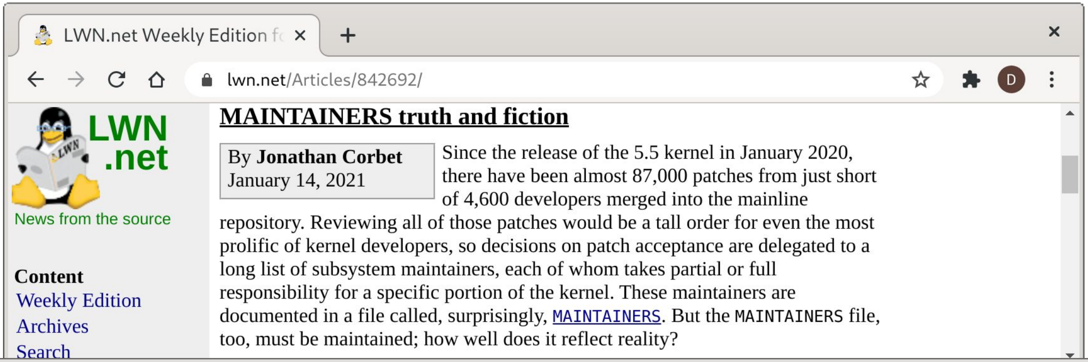
Since the release of the 5.5 kernel in January 2020, there have been almost 87,000 patches from just short of 4,600 developers merged into the mainline repository. Reviewing all of those patches would be a tall order for even the most prolific of kernel developers, so decisions on patch acceptance are delegated to a long list of subsystem maintainers, each of whom takes partial or full responsibility for a specific portion of the kernel. These maintainers are documented in a file called, surprisingly, `MAINTAINERS`. But the `MAINTAINERS` file, too, must be maintained; how well does it reflect reality?



Some `MAINTAINERS` entries have broad lists of covered files that make the commit count seem larger than it really is. For example, the subsystem named "ASYNCHRONOUS TRANSFERS/TRANSFORMS (IOAT) API" includes all of `drivers/dma`, which is also claimed by "DMA GENERIC OFFLOAD ENGINE SUBSYSTEM". That subsystem, in turn, is actively maintained by Vinod Koul. There are two subsystems that fall into this category; in the tables below "Activity" indicates the last observed activity by the listed maintainers (if any), while "Commits" shows the number of commits affecting the subsystem since 5.5:

| Subsystem | Activity | Commits |
|---|---|---|
| ASYNCHRONOUS TRANSFERS/TRANSFORMS (IOAT) API | —— | 536 |
| HISILICON NETWORK SUBSYSTEM DRIVER | 2019-11-16 | 258 |

LF live MENTORSHIP SERIES


LWN.net Weekly Edition f

lwn.net/Articles/842692/

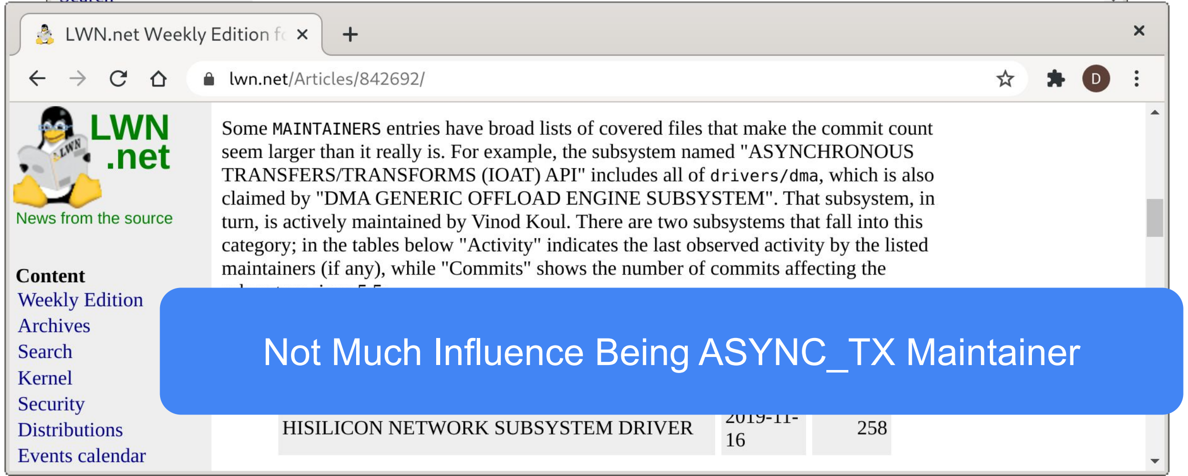
LWN .net News from the source

Content
Weekly Edition
Archives
Search

**MAINTAINERS truth and fiction**

By **Jonathan Corbet**
January 14, 2021

Since the release of the 5.5 kernel in January 2020, there have been almost 87,000 patches from just short of 4,600 developers merged into the mainline repository. Reviewing all of those patches would be a tall order for even the most prolific of kernel developers, so decisions on patch acceptance are delegated to a long list of subsystem maintainers, each of whom takes partial or full responsibility for a specific portion of the kernel. These maintainers are documented in a file called, surprisingly, `MAINTAINERS`. But the `MAINTAINERS` file, too, must be maintained; how well does it reflect reality?


LWN.net Weekly Edition f

lwn.net/Articles/842692/

LWN .net News from the source

Content
Weekly Edition
Archives
Search
Kernel
Security
Distributions
Events calendar

Some `MAINTAINERS` entries have broad lists of covered files that make the commit count seem larger than it really is. For example, the subsystem named "ASYNCHRONOUS TRANSFERS/TRANSFORMS (IOAT) API" includes all of `drivers/dma`, which is also claimed by "DMA GENERIC OFFLOAD ENGINE SUBSYSTEM". That subsystem, in turn, is actively maintained by Vinod Koul. There are two subsystems that fall into this category; in the tables below "Activity" indicates the last observed activity by the listed maintainers (if any), while "Commits" shows the number of commits affecting the

| | | |
|---|---|---|
| HISILICON NETWORK SUBSYSTEM DRIVER | 2019-11-16 | 258 |

## Not Much Influence Being ASYNC_TX Maintainer

## How to Kernel: Review

- Be a connoisseur and collector of mistakes
  - Read the docs

## How to Kernel: Review

- Be a connoisseur and collector of mistakes
  - Read the docs
  - Mistakes landing your feature (you read the docs, right?)

## How to Kernel: Review

- Be a connoisseur and collector of mistakes
  - Read the docs
  - Mistakes landing your feature (you read the docs, right?)
  - The mistakes that veteran reviewers find

    f:torvalds s:"GIT PULL"

## How to Kernel: Review

- Be a connoisseur and collector of mistakes
  - Read the docs
  - Mistakes landing your feature (you read the docs, right?)
  - The mistakes that veteran reviewers find

    f:torvalds s:"GIT PULL"
  - Do not repeat them

## How to Kernel: Review

- Be a connoisseur and collector of mistakes
  - Read the docs
  - Mistakes landing your feature (you read the docs, right?)
  - The mistakes that veteran reviewers find

    f:torvalds s:"GIT PULL"
  - Do not repeat them
  - Bonus points: preclude mistakes

## How to Kernel: Review

“...if you are starting from the premise that ‘I don't know this code well enough to perform a useful review’ then you are setting yourself up for failure right at the start. Read the series description, think about the change being made, use your experience to answer the question ‘what's a mistake I could make performing this change’. Then go looking for that mistake through the patch(es). In the process of performing this review, more than likely, you'll notice bugs other than what you are actually looking for…”

- David Chinner

## How to Kernel: Review

"...if you are starting from the premise that 'I don't know this code well enough to perform a useful review' then you are setting yourself up for failure right at the start. Read the series description, think about the change being made, use your experience to answer the question 'what's a mistake I could make performing this change'. Then go looking for that mistake through the patch(es). In the process of performing this review, more than likely, you'll notice bugs other than what you are actually looking for…"

- David Chinner

Opportunity to change the core is rare, opportunity to review core changes is abundant

## How to Kernel: Review

- Subject matter expertise not required
  - Leverage knowledge of one subsystem to gain a foothold in the next
  - Use naivete to your advantage (changelogs and documentation)

## How to Kernel: Review

- Seek and destroy contributions to the Platform Problem
  - Debt collects where developers avoid the platform
  - Platform pain may take review to become apparent
  - Red diffs are pure candy

## How to Kernel: Review

- Influence is granted to those that demonstrate they are actively carrying the burden of keeping the kernel healthy and relevant.
- Negotiate Review time with your manager (rebuild trust capital to spend on your next mistake)

LF Live MENTORSHIP SERIES

## How to Kernel: Maintain

## How to Kernel: Maintain

- What does success look like?

  >10 years of active participation

  Reviews trusted by many maintainers

  No entry in MAINTAINERS

The goal is be a reliever of burdens, not a target of burdens

## How to Kernel: Maintain

On Sat, Jun 9, 2018 at 8:17 AM Dan Williams <dan.j.williams@intel.com> wrote:
>
> Well, crap. I've been doing it the wrong way for a while.

... you're not the only one ;*(

I only really catch it when it's very obvious, like it was now when
the last merge was just before so it stood out like a sore thumb when
I looked at the resulting "git log".

... or then I catch it in the (happily rare) case when a merge causes
issues, and then I curse.

## How to Kernel: Maintain

- Towards a Maintainer Handbook?
- Every subsystem is different
- Seek mentoring
- Declare expectations: Subsystem Profile
- Be predictable

## Why?

## Why? / Conclusion

- The Influence, Trust, Impact Cycle
- The project needs you to persevere
- Just for fun

## Why? / Conclusion

- The Influence, Trust, Impact Cycle
- The project needs you to persevere
- Just for fun

...but seriously, go review some patches!

LF live MENTORSHIP SERIES

## Thank you for joining us today!

We hope it will be helpful in your journey to learning more about effective and productive participation in open source projects. We will leave you with a few additional resources for your continued learning:

- The LF Mentoring Program is designed to help new developers with necessary skills and resources to experiment, learn and contribute effectively to open source communities.
- Outreachy remote internships program supports diversity in open source and free software
- Linux Foundation Training offers a wide range of free courses, webinars, tutorials and publications to help you explore the open source technology landscape.
- Linux Foundation Events also provide educational content across a range of skill levels and topics, as well as the chance to meet others in the community, to collaborate, exchange ideas, expand job opportunities and more. You can find all events at events.linuxfoundation.org.