

Netdev Processes, Community and Respect

Andrew Lunn <andrew@lunn.ch>

July 16, 2024

- 1 Introduction
- 2 Submitting Patches
- 3 Maintainer
- 4 Reviewing Patches

Section 1

Introduction

Discussion, not death by Beamer¹

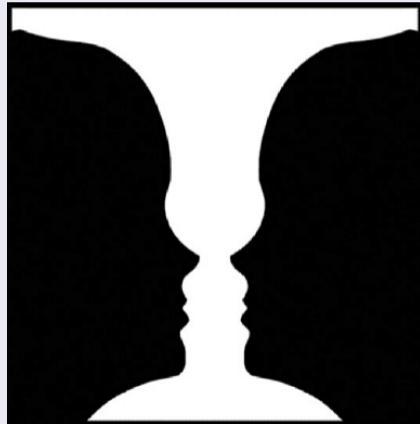
- Inspired by some questions at 0x17, LPC, LWN article about Rust PHY driver, and chatting with community members
- This should be a chance for the Community to discuss how Netdev works
- Ask questions, raise problems, we can try to fix things which are broken
- I don't care if it don't get to the last slide

¹Beamer is a \LaTeX package for slides, akin to Power Point

This is my perception of how netdev works

- Some of this is written down
- Some is {maybe} agreed tribal knowledge
- Some is just how i see it
- If you disagree, say so... that in itself is a useful discussion

Two faces? Vase?



Section 2

Submitting Patches

What is written down:

- <https://docs.kernel.org/process/submitting-patches.html>
- <https://docs.kernel.org/process/maintainer-netdev.html>
- <https://docs.kernel.org/process/stable-kernel-rules.html>

- Designate your patch to a tree - [PATCH net] or [PATCH net-next]
- For fixes the Fixes: tag is required, regardless of the tree
 - CC: stable@vger.kernel.org
- Don't post large series (> 15 patches), break them up
 - The bigger the series, the less likely it will be reviewed
- Don't repost your patches within one 24h period
 - Waste reviewers time reviewing old versions
- Reverse xmas tree

Patchworks, patch status, CI & Testing

- <https://patchwork.kernel.org/project/netdevbpf/list/>

Netdev + BPF - Patch

patchwork.kernel.org/project/netdevbpf/list/?series=&submitter=3921&state=*&q=&archive=&delegate=

Patchwork Netdev + BPF Patches Bundles About this project Login Register Mail settings

Show patches with: Submitter = Andrew Lunn | Archived = No | 328 patches

Patch	Series	A/R/T	S/W/F	▲ Date	Submitter	Delegate	State
[net-next] net: genet: Fixup EEE	[net-next] net: genet: Fixup EEE	---	19 - -	2024-04-09	Andrew Lunn	netdev	Changes Requested
[net-next,2/2] net: lan743x: Fixup EEE	net: Clean up some EEE code	- 1 -	19 - -	2024-04-06	Andrew Lunn	netdev	Accepted
[net-next,1/2] net: usb: lan78xx: Fixup EEE	net: Clean up some EEE code	- 1 -	18 1 -	2024-04-06	Andrew Lunn	netdev	Accepted
[net-next,v4,8/8] arm: boot: dts: mvebu: linksys-mamba: Add Ethernet LEDs	net: Add generic support for netdev LEDs	---	17 1 -	2024-04-06	Andrew Lunn	netdev	Changes Requested
[net-next,v4,7/8] dsa: mv88e6xxx: Create port/netdev LEDs	net: Add generic support for netdev LEDs	---	16 - 2	2024-04-06	Andrew Lunn	netdev	Changes Requested
[net-next,v4,6/8] net: dsa: mv88e6xxx:	net: Add generic support for netdev				Andrew		Changes

NIPA - Netdev Infrastructure for Patch Automation

<https://github.com/linux-netdev/nipa>

The screenshot shows a web browser window displaying a patchwork page. The browser's address bar shows the URL: `patchwork.kernel.org/project/netdevbpf/patch/20240406-v6-8-0-net-next-mv88e6xxx-leds-v4-v4-7-eb97665e7f96@lunn.ch/`. The page header includes navigation links for 'Patchwork', 'Netdev + BPF', 'Patches', 'Bundles', and 'About this project', along with 'Login', 'Register', and 'Mail settings' options. The main heading of the patch is '[net-next,v4,7/8] dsa: mv88e6xxx: Create port/netdev LEDs'. To the right of the heading are buttons for '13619923', 'diff', 'mbox', and 'series'. Below the heading, the 'Message ID' is '20240406-v6-8-0-net-next-mv88e6xxx-leds-v4-v4-7-eb97665e7f96@lunn.ch' with a link to the 'mailing list archive'. The 'State' is 'Changes Requested', 'Delegated to' is 'Netdev Maintainers', 'Headers' has a 'show' link, and 'Series' has a link to 'net: Add generic support for netdev LEDs | expand'. A 'Checks' section follows, containing a table with columns 'Context', 'Check', and 'Description'. The table lists five checks, all of which are successful.

Context	Check	Description
netdev/series_format	success	Posting correctly formatted
netdev/tree_selection	success	Clearly marked for net-next, async
netdev/ynl	success	Generated files up to date; no warnings/errors; no diff in generated;
netdev/fixes_present	success	Fixes tag not required for -next series
netdev/header_inline	success	No static functions without inline keyword in header files

- Patchwork / NIPA intended for Maintainers
- You will get SHOUTED at if you abuse it
- Do not post patches just so they get tested

- Netdev moves fast, faster than pretty much every other subsystem
- Patches will be triaged in about three working days
- If there are no comments, patches will get merged
 - But somebody always looks at the patches, NIPA, etc.
- A respin of a patchset every week is reasonable

"Release early. Release often. And listen to your customers"²

²The Cathedral and the Bazaar, Eric S. Raymond

Expectations from the Community (That's You)

- Read and follow the usual basics
 - <https://docs.kernel.org/process/submitting-patches.html>
 - <https://docs.kernel.org/process/maintainer-netdev.html>
 - <https://docs.kernel.org/process/stable-kernel-rules.html>
- Code meets the usual quality expectations, W=1 C=1, etc.
- Code has been build tested similar to Patchwork/NAPI
- Ideally, include some tests for new features

Expectations from the Community (That's You)

- Read the netdev mailing list and search the archive
 - Learn from others mistakes and successes
 - Great source of tribal knowledge
 - Don't submit something already rejected
 - It is not expected you read everything.
 - Keep an eye out for what is relevant to you

Expectations from the Community (That's You)

- Community will help 'New hackers in their basement hacking for fun'
- Community expects 'professional hackers' to submit high quality patches
 - The bigger the company, the better the quality
 - Big companies are expected to train and mentor their employees
 - Vendor Crap hackers need training and time to become Mainline coders
 - PHB need as much training about mainline as developers
 - Community has no time/desire/benefit to train 'professional hackers'

Expectations from the Community (That's You)

- Are these reasonable expectations for submitting patches?
- Are there other expectations for patches?
- Any expectation you as the community think we should add?

Submissions and Respect and Reputation

- Submitters start with a neutral respect level
- Good patches increases the communities respect level of developers and company
- Bad patches decreases the communities respect level of developers and company
- Good respect levels makes getting patches merged easier and faster
- Bad respect levels will increase scrutiny

Submissions and Respect and Reputation - For a Company

- One bad submitter can damage a companies respect/reputation level
 - Can also damage respect across subsystem boundaries
 - SoC develops: Don't upset Device Tree Maintainers
- Community does not really see subsidiaries within a company
- The experience developers in a company should jump in and mentor the newbies

From: Przemek Kitszel <przemyslaw.kitszel@intel.com>

I'm welcoming you to post next version of the driver to the IWL mailing list, and before that, to go through our Intel path for ethernet subsystem (rdma and a few smaller ones also go through that) (that starts internally, I will PM you the details)

Section 3

Maintainer

Become a Maintainer by being a Maintainer

- Need to be respected by the community
 - Show you know the code
 - Show you know the processes
 - Show you know the Community
- Respect can be lost by being absent
 - Eventually purged from MAINTAINERS
- It is the Community who decides, not a Vendor
- Most become Maintainers via a driver
- Also possible to review lots of patches

Expectations on a Maintainer

- Review patches submitted in your domain
- Guide architectural discussions
 - You don't need to do all the work, point others in the right direction
- Take part in the community
- Nice to let others know about vacations etc.

Expectations from the Community for Maintainers (That's You)

- Are these reasonable expectations for Maintainers?
- Are there other expectations for Maintainers?
- Any expectation you as the community think we should add?

Section 4

Reviewing Patches

Reviewing patches is about Quality Assurance and avoiding future pain

- Patches need reviewing for QA reasons
 - Find bugs early
 - Fit the architecture?
 - Not "Vendor Crap"
- Getting kAPIs right is hard
- Enforce our community goals/spirit/ideals

Reviewing patches has to be done by the Community

- Does not scale without community involvement

Top scores (positive):

```
1 (  ) [500] RedHat
2 (  ) [359] Meta
3 (  ) [176] Andrew Lunn
4 ( +1) [110] Google
5 ( +1) [ 70] Enfabrica
6 ( +1) [ 53] Oracle
7 ( -3) [ 47] Linaro
```

Top scores (negative):

```
1 ( +1) [86] Alibaba
2 (+12) [65] Huawei
3 ( +5) [58] Intel
4 (**) [55] Dent Project
5 (+45) [44] Asbjørn Sloth Tønnesen
6 ( -5) [38] Bootlin
```

Shows Reviews vs Patches³

Tries to show who is doing the work and who is skipping their fair share of the work.

³Stats for 6.10 cycle <https://lore.kernel.org/netdev/20240515122552.34af8692@kernel.org/>

Get out of your Silo

- Many developers have no idea what others similar drivers are doing
- Don't know what great ideas they can copy into their code
- Common code which will make the next driver easier to write
- Reviewing other drivers will make your own drivers better

Consider it payment for being part of the community

- You are getting your code improved for free
- Others are helping maintaining it
- You get to influence the direction the community takes
- Your own patches are probably better as a result
- Educate the PHB that doing reviews is a cheap price to pay!

- Netdev moves fast, faster than pretty much every other subsystem
- Ideally reviews happen within three working days
 - If you want to review, but don't have time, let people know
- If you are too slow you can still point out bugs after code has been merged
- Try to review each revision of a patch until you give it a Reviewed-by:

Reviewer and Respect/Reputation

- Reviewers start out with neutral respect level
 - Newbies to reviewing won't harm submissions
- Gain respect by making useful comments
 - Point out code quality issues
 - Process issues
 - Community issues
- Reviewed-by: alone does not gain respect
- Reviewed-by: from same organisation has little value
- Negative respect is possible and comments will be ignored

What reviewer get wrong

- Trim replies to just the needed context
 - Comments get lost in a 1000 line reply.
- It would be nice if somebody wrote a bot SHOUTING at people who don't trim replies

Tips and Tricks for Reviewing

- email client matters: Mutt, alpine, emacs etc.
- email client which is efficient at deleting email
 - You will delete 95% without reading it
- email client should fork \$EDITOR
 - Reviewing code is not too different to writing code
 - Make use of your muscle memory for jumping around, searching etc.
 - Syntax highlighting in colour etc.
- Define macros to do common things:
 - Reviewed-by:
 - pw-bot: cr
 - form responses

Getting Started

- No need to review a whole driver
- Maybe pick a different market segment
 - Automotive Ethernet not that different to data centre
 - Everybody gets pause wrong, independent of market segment
- Maybe pick one thing and review just that within all drivers
 - Firmware mailbox APIs
 - DMA code
 - Statistics
- Ask questions if you don't feel confident to say it is wrong

Expectations from the Community (That's You)

- Are these reasonable expectations for reviewing?
- Are there other expectations on reviewers of patches?
- Any expectation you as the community think we should add?

Questions?

Any questions or comments about:

- processes
- community
- respect
- 42