Trying to Explain the "Incomprehensible" Decision Making Process of a Subsystem Maintainer

Wolfram Sang

Consultant / Renesas Upstream Kernel Team

Overview

1. The root cause
2. What does that mean?
3. What you can do (or shouldn’t do)
4. Changes to I2C subsystem workflow
Overview

1. The root cause

2. What does that mean?

3. What you can do (or shouldn’t do)

4. Changes to I2C subsystem workflow
What is a maintainer? Trond Myklebust said it nicely¹:

"Currently, the Linux maintainer appears to be responsible for filling all of the traditional roles of:

- software architect
- software developer
- patch reviewer
- patch committer
- and software maintainer

I think at least "educator" needs to be added, too.

¹link here
Statistics: # of patches

NumCommits

Wolfram Sang (Consultant)

Trying to explain maintainers
Merges not counted
But they are work, too

Stats only based on accepted patches
There are also superseded and rejected patches, teaching new authors...

Situation at v3.0 was already far from ideal
Statistics: # of tags (linearized)
Statistics: # of people

The graph shows the number of authors and committers over the years from 2011 to 2016. The number of authors has been relatively stable, while the number of committers shows a slight increase over the years. The graph also includes data points for versions v3.0 to v4.5.
Wolfram Sang (Consultant)  Trying to explain maintainers  13.10.2016, ELCE2016  12 / 31
Statistics: more # of people (mostly linearized)
Unprocessed patches (up to now)

Wolfram Sang (Consultant)
Trying to explain maintainers
13.10.2016, ELCE2016
Unprocessed patches (skipping last cycles)
Unprocessed patches (normalized): a trend?
I dare to disagree

LWN on May 11th, 2016

Quote:
"The overall picture ... is one of a development process that continues to function like a relatively well-tuned machine. The number of contributors continues to increase, the patch flow is steady, and there do not appear to be many process-scalability issues in sight."

- I think there is a scalability problem
- I am neither a machine nor part of a machine
  actually I am full of human factors ;)

Wolfram Sang (Consultant)
Trying to explain maintainers
13.10.2016, ELCE2016
Overview

1. The root cause
2. What does that mean?
3. What you can do (or shouldn’t do)
4. Changes to I2C subsystem workflow
In what order patches are processed?

Factors affecting when a patch is processed

- You help me, I help you
  typical human factor, I’d assume
- Kernel-wide or cross-subsystem effort
  if I2C is only one part of it
- number of affected users
- regression?
In what order patches are processed?

Factors affecting when a patch is processed

- complexity
- polished or not
- chronological order
  - a lot less important than I’d like to
- new driver?
  - rc1 rule might apply
Things mainly maintainers care about

- removal of obsolete features from 2.4 times
  largely means messing with PowerMac drivers
- refactoring the I2C core to ease maintenance
- split up the core into parts which can be maintained separately
- give users better testing tools
- update documentation & wiki page

Most of that is currently delayed for years!
Overview

1. The root cause
2. What does that mean?
3. What you can do (or shouldn’t do)
4. Changes to I2C subsystem workflow
What you can do: Users²

Give feedback

- give comments about patches
  show interest, tell about issues, ...

- give tags
  Tested-by! Very important one, no need to be a coder for that

²as in ”users of patches”
Always give your best shot

- missing experience is no problem
- sloppiness is a problem
- be honest, give reasons for suboptimal solutions
- for companies, look for in-house knowledge

have your tools ready

- identify repetitive tasks, automatize them
  - Keyboard shortcuts!
- run (& understand) those code checkers! Always!
  - checkpatch, sparse, smatch, cocciscript
What you can do: Developers

review your own patches

- don’t just send a ping
- If you didn’t touch the patch for a while and have some distance, you are a potential reviewer as well
- → big credit boost

take part in further reviewing

- review
- discuss
- clean up, consolidate
- if you are interested, become a (sub-)maintainer
What you shouldn’t do

ping considered harmful

- "ping after 2 weeks" is outdated
  "ping after 2 month" would be closer to reality

- largely not needed
  all people I know have patch tracking systems in place

- I won’t reply anyway
  I could review patches in that time...

- Human factor: they still add to frustration
  although I do know the latency is not my fault

- Private pings are especially bad
  reviewing should be a community effort

- If you still need to ping, try to think if you can help somehow
Overview

1. The root cause
2. What does that mean?
3. What you can do (or shouldn’t do)
4. Changes to I2C subsystem workflow
What is a maintainer? Trond Myklebust said it nicely:\[3\]:

"Currently, the Linux maintainer appears to be responsible for filling all of the traditional roles of:

- software architect
- software developer
- patch reviewer
- patch committer
- and software maintainer

"educator"

\[3\]link here
What is a maintainer\(^4\) in the future?

- **software architect**
  one of the software architects
- **software developer**
  one of the software developers
- **patch reviewer**
  one of the patch reviewers
- **patch committer**
- **software maintainer**
- (new focus!) advertiser for distributed community efforts

\(^4\)well, at least the I2C maintainer
Changes to I2C

Disadvantages

- expect bigger latencies

Advantages

- I keep sane
  ...because I am neither a machine nor part of a machine
- can spend more time fixing this issue on higher levels
Thank you for your attention!

Let’s work together

Questions?
- Right here, right now…
- Later at the conference
- wsa@the-dreams.de

Breaking news
- at the GPL BoF lunch today
  meeting at the lobby