LF live

MENTORSHIP SERIES

LF Live MENTORSHIP SERIES

Which hat do I wear today?

Shuah Khan, Kernel Maintainer and Fellow,
The Linux Foundation

Do you find yourself balancing product needs with open source project objectives?

You are not alone ...

Navigating open source and corporate cultural differences ...

- Delicate and challenging balancing act
- Easier to work exclusively on either side of this divide
- Requires finesse and skill to know when to wear which hat
- To be productive and bring value to both sides

Let’s talk about ...

- Upstream mindset
- Doing your homework
- Balancing act
- Bridging between two worlds
- Final thoughts
- Q&A

Upstream mindset ...

- Upstream releases are time based - not feature based
  - New release once every 8-10 weeks
  - 2 weeks merge window for new features
  - 6-8 weeks for fixing bugs and regressions
  - All fixes must go into the mainline first
  - Working with the kernel development community

Upstream mindset ...

- Fixes going into the mainline get applied to stable releases
  - Selection fixes for stables is automated
  - Attempt to apply automatically first
  - If auto apply fails, porting is requested
  - Manual selection is necessary for some fixes
  - Can be marked for stable using “for stable” tags
  - Everything you ever wanted to know about Linux -stable releases

Upstream mindset ...

- Some stable releases are tagged as LTS
    - longer term updates for fixes
- Upstream doesn't care about your product needs
    - deadlines
    - time to market needs
    - features
- Upstream cares about features that add value

Doing your homework ...

- Research related upstream features
  - Very important first step
- Leverage and enhance existing features
  - Benefits all
- New features working well within the framework
  - Very important to design new features to work with the existing framework

Doing your homework ...

- Why are your making the change?
- Are you proposing changes to an existing feature?
- Are you proposing a new feature?
- What’s the best way to address product needs within the time constraints
    - Working upstream requires planning
    - Can take longer to get feature in
    - Benefits come later in the form of collaboration

Upstream hat on ...

- Is the solution/feature generic for upstream?
- How does your work help upstream?
  - Does it add new functionality
  - Does it improve performance/security?
  - Does it harden the code base?
  - Is there a better way to do it?
- Would you take this solution if it comes from another developer?

Product hat on ...

- Upstream doesn't work on your platform ...
- Hardware/firmware deviates from standard …
- New feature or enhancements an existing one anchors your product
- Can you talk about your hardware/firmware differences?
- How can you make a case without giving the store away?
    - Find a way to discuss the details in a generic way

Balancing act ...

- Don’t expect upstream to take your solution as is
  - Discuss with upstream at conferences
  - Collaborate and share early
- Adapt upstream first rule (best policy in the long run)
  - Upstream first and then into the product (best option)

Balancing act ...

- Keep upstream variations to a small %
    - Decide how often to merge upstream (frequent is better)
- Keep track of Stable and LTS releases
    - make sure your content hits the target
    - Contingency plan if it doesn't

Bridge between two worlds ...

- Explaining upstream to product teams
  - Benefits and importance of upstream first
  - Paying it forward for benefits of collaboration with upstream
  - keeping delta from upstream small
- Explaining your feature or change to the upstream community
  - Getting it accepted by the upstream community
- Communicate “What’s in it for you message to both sides”

Final thoughts ...

- Be open to changing your solution
    - Assume your **idea/solution** will change and evolve
    - Don’t be locked into it
- Be mindful of adding value to both worlds
- Be open to be a student and teacher
- Leverage collective expertise of the community
    - Attitude of “I don’t know everything”

Learn as you go

Keep learning

There is no text book that teaches these skills

MENTORSHIP SERIES

Questions - comments - thoughts

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.