Subject: Upstream release dependency
Posted by rk4n3 on Wed, 28 Aug 2019 08:07:22 GMT
View Forum Message <> Reply to Message

Hi all,

In considering the recent post concerning the retirement of ConnochaetOS, I noticed that one cited factor was the lag in upstream release.

As we're all aware, a relatively unprecedented period of time between releases has us waiting for our next upstream. I'm curious about anyone's thoughts on how this should be handled moving forward.

The most obvious option would be to merely take what we're given, exercising patience/tolerance for any lag/delay.

I could certainly accept this option as practice, if it turns out to be "the best option" per deliberations. Still, I'm curious about whether anyone thinks other options might be better

---

Subject: Re: Upstream release dependency
Posted by rk4n3 on Wed, 28 Aug 2019 10:08:20 GMT
View Forum Message <> Reply to Message

Here's just one example of an idea ...
We could do something similar to some other distros:

 Have an LTS (Long-Term Support) release based on upstream's stable releases

 Have a semi-rolling snapshot release based on upstream --current, perhaps semiannually

So, freenix 14.2 would be the latest LTS, the next LTS would be based on the next upstream stable release, and then we could start releasing semiannual snapshot releases based on --current at the time of release, and named like freenix 15.0.1, 15.0.2, etc.

If upstream goes another 3+ years without a stable release, freenix would have roughly 6 releases, keeping up with --current, while retaining the option for people to stick with the LTS if they want to.

Subject: Re: Upstream release dependency
Posted by connie on Wed, 28 Aug 2019 19:17:40 GMT
View Forum Message <> Reply to Message

I think it's a great idea, but it would definitely require us to build all new infrastructure, so that thing would have to have its own repo. That's a lot of extra work, but something to be considered at times like now, when theres's a significant demand for updated packages.

Another approach would be to identify the specific packages in -current with a lot of user demand (what are they today?), and push them into FXP, letting users basically stay with stable branch, and use FXP like a -current backports repo. In fact, I already did that once with the kernel, when Pat lagged with bugfix for some reason. Not only this requires like 100 times less work, but it also gives users the option of one-touch reverting to stable state, in case things go sour with new packages. Downgrading the entire OS is probably quite a bit nastier.