

### **Original version**

Posté par [patrick\\_g \(site web personnel\)](#) le 05/07/11 à 11:01. En réponse à la dépêche [Un entretien avec Lennart Poettering](#). Évalué à 10. Dernière modification le 06/07/11 à 06:37.

**LinuxFr.org : First Avahi then PulseAudio and now systemd. You seem to be very versatile isn't it ? How is it possible to contribute like this in different and unrelated fields ?**

**Lennart :** Well, they are more related than people might think. For example, I ended up writing Avahi when I was looking for a nice way to discover audio devices on the network while hacking on PulseAudio. I learned about mDNS/DNS-SD (i.e. the network discovery protocol stack Avahi implements), and back then an implementation called Howl was the most used mDNS/DNS-SD implementation, so I had a closer look on it hoping to add support for it in PulseAudio. However, I quickly noticed its shortcomings and its broken licensing, and after reading the RFCs of mDNS/DNS-SD I decided this was simple enough to implement and I could do a clean reimplementation in a few days. (It wasn't quite that simple after all, but that's another story...)

When working on Avahi I learned a lot about the complexities of safely and reliably running and maintaining system services, and about securing them as much as possible, which is particularly important for network facing services like Avahi. I implemented a lot of pretty nifty features in this area in Avahi. For example, Avahi is still pretty much the only daemon on a standard Linux install that chroot(s) itself by default. (Well, there's another one now, the RealtimeKit daemon -- but I wrote that one too...). Also, it was one of the first daemons around which dropped process capabilities by default.

Back then I first started to think about how an init systemd should really handle all these things, so that the complexities needed to implement things like this go away. And yeah, I always kept thinking about this, and then ultimately Kay and I decided we really needed to something about this, and finally we did.

**LinuxFr.org : During the early days of PulseAudio there was a lot of criticisms. Do you think PA deserved them because it was a bit immature or do you think the criticisms were irrational and/or misguided ?**

**Lennart :** I can understand why people were upset, but quite frankly we didn't really have another option than to push it into the distributions when we did. While PulseAudio certainly wasn't bug-free when the distributions picked it up the majority of issues were actually not in PulseAudio itself but simply in the audio drivers. PulseAudio's timer-based scheduling requires correct timing information supplied by the audio driver, and back then the drivers weren't really providing that. And that not because the drivers were really broken, but more because the hardware was, and the drivers just lacked the right set of work-arounds, quirks and fixes to compensate for it.

You should never forget that in the whole industry there are about 3.5 people paid full-time for doing generic maintenance work of the Linux audio stack (which I consider consisting primarily of ALSA and PulseAudio and a few things around it). With this little manpower I can only say that what has been achieved is pretty good. While we still can't fully match competing audio stacks like CoreAudio, we are a lot closer than we ever were. I do hope that the folks who kept constantly complaining would be a lot more appreciative if they understood that.

Most of this is a think of the past. Hardware vendors these days tend to understand that when PulseAudio doesn't work properly on their hardware it's most likely their fault, not PulseAudio's, and to some extent PulseAudio has advanced to become a standardized test hardware vendors check their drivers against. Of course, especially on new hardware you might still run into pro-

blems, simply because even though there are somewhat standardized specifications for sound card designs like HDA or USB Audio hardware designers always find new and creative ways to implement them badly and incorrectly.

Also, what other option would there have been? It's pretty naive to believe that if we had waited any longer with pushing PulseAudio into the distributions things would have gone any different: you cannot fix bugs you don't know about, and the incentive and manpower is too small to get them fixed without having the pressure that the stuff is shipped.

And finally, CoreAudio on MacOS required a few iterations to get everything into shape too. Nowadays CoreAudio is pretty good as a benchmark for audio stacks, but you'll find a lot of folks who will tell you how it was when it was first introduced, and those aren't stories full of love. (Oh, and it still isn't all unicorns and rainbows yet, the last time when I tried to combine a USB with a built-in audio card on MacOS the kernel didn't really like that so much). An audio stack that can do timer based scheduling is necessarily complex, and hence difficult to get right. I am pretty sure most folks will notice that when they try to implement one and need to while to fully stabilize it.

**LinuxFr.org : If you compare the audio landscape in MacOSX or Windows and the situation in Linux do you think we are on par with the proprietary systems ?**

**Lennart :** Both Windows and MacOS have much better integrated audio stacks than we have. We have a couple of unique features they can't match, like networking support, lower latency behaviour, or float-less audio pipelines, but in general the CoreAudio stack is definitely more advanced than ours. The distance, however, is much smaller than it used to be.

The Windows stack is much less suitable as a benchmark, and we are matching its features much better than CoreAudio's, but there's no doubt that it is probably still a better integrated audio hacking experience to develop for Windows.

**LinuxFr.org : What are the advantages of ALSA/PA versus OSSv4 ? Why do you think the BSDs still use OSS instead of reimplementing ALSA/PA ?**

**Lennart :** OSS is a simplistic 90's style audio stack. I doesn't really have any relevance for what you need for a modern desktop. But that's fine, the BSDs don't focus on the desktop, and on the server you don't need any kind of audio. Also, and this key: one of the reasons the BSDs and Solaris are stuck with OSS is that they don't really have any other options.

The OSS model is too simple to expose properly what modern hardware is like. You cannot implement logic like timer-based scheduling on it (which is mandatory to properly handle more than one client with different latency constraints or latency at all, and all that in a power consumption friendly way), and doing mixing and sample conversion in the kernel is pretty questionable too. The mixer interface is simplistic, and not useful really for modern devices.

Audio devices have changed, you nowadays have Bluetooth audio, UPnP audio, USB audio, and other complex audio technology. The OSS model was pretty much designed for cards like the classic SoundBlaster series, but having the device node as the primary interface to the audio devices is not really suitable if you are interested in the more advanced technologies.

**LinuxFr.org : OpenBSD use the auct sound server ([http://openbsd.das.ufsc.br/papers/asiabsdcon2010\\_sndio.pdf](http://openbsd.das.ufsc.br/papers/asiabsdcon2010_sndio.pdf)). Do you know it ? What do you think about this server ? More generally are you interested by the other systems in order to discover new ideas ?**

**Lennart :** An audio stack that is not capable of timer-based scheduling and dynamic latency control based on that is not useful on consumer devices. To my knowledge PulseAudio is the only free implementation of that design which was originally made popular by CoreAudio. Also, a zero-copy design like the one PulseAudio has (where you exchange references to audio data in memory instead of audio data itself between applications and the stack) is pretty much unmat-

ched too.

The irony is that audio stacks for pro audio can generally be implemented much simpler than those for consumer audio and mobile devices. On pro audio power consumption is irrelevant, and latency is key. Due to that latency control becomes very simple (you just want it as low as possible) and technologies like zero-copy are not interesting (since audio frames become as short as the references they could be described in). Coming up with an audio stack that was useful for consumer PCM uses is hence much harder than one for pro audio PCM. And any audio stack not implementing timer-based scheduling or zero-copy design is simply of very little relevance to consumer devices. It's the same reason why Jack and PulseAudio don't really compete.

All free audio stacks (except PA+ALSA) I have seen so far do not cover timer-based scheduling or zero-copy data transfer at all. And as long as that is the case you'll always compare apples and oranges when you wonder whether projects like the OpenBSD audio stack or PulseAudio are better.

**LinuxFr.org : Except Ubuntu it seems that systemd will be the future init system for most of the major distributions. Did you expect this quick success ?**

**Lennart :** We quickly knew we had created something really good, but it was much easier to push this through than we anticipated. It took as one year from announcement to get it shipped in f15, and afaics it has been quite successful in it. In many ways it was like running along a long corridor and kicking in open doors.

And I am pretty sure we'll eventually make it into Ubuntu too. Already Upstart can't really keep up with systemd, and I figure Canonical will notice that one day too. Since Scott stepped down as maintainer the project lost its visionary, so I figure they'll eventually notice that Upstart is leading nowhere, and as the gap widens they'll eventually let it go. -- But hey, that's just my guess, and I don't work for Canonical, so I don't really know and only time will tell. Maybe it's all just wishful thinking on my side...

**LinuxFr.org : Besides being a great hacker you are able to promote systemd very well with detailed posts on your blog. Do you think it's part of your job to do this "communication" task ?**

**Lennart :** It is. We are supposed to blog about our work. It's one of the great things working for Red Hat: they encourage you to be open and blog about the tech you work on.

**LinuxFr.org : What are the technical advantages of systemd versus upstart ? Do you think Canonical will change the Ubuntu's init system eventually ?**

**Lennart :** There are so many technical advantages of systemd vs. Upstart! I think [this blog story I posted a while back](#) tells it all.

And yupp, I think they'll eventually switch too, see above. But I can only speculate.

**LinuxFr.org : Could explain a little bit your sentence: "Systemd is the first Linux init system which allows you to properly kill a service" ?**

**Lennart :** A service like Apache might have a number of subprocesses running, like CGI scripts, which might have been written by 3rd parties and hence are very lightly controlled only. This gives them the power to detach themselves almost completely from the main Apache service, and this actually happens in real life.

In systemd this is not possible anymore, and processes can no longer escape the supervision. That enables us -- for the first time -- to fully kill a service and all its helper processes, in a way that we can be sure no CGI script can escape us.

For details see [this blog story](#) I posted a while back.

It's kinda ironic that the job of killing a service which appears to simple at first is actually quite

hard to get right, and only now we could fix this properly. One might have hoped this issue would have been fixed much earlier on Linux.

**LinuxFr.org : I heard that you plan to use systemd to replace and/or extend the existing desktop sessions managers. Could you elaborate ? What are the selling points of systemd for this task ?**

**Lennart :** The task of starting a number of services quickly and reliably is not unique to system boot-up. Pretty much the same is done when a user logs in and the user services are spawned. The same technologies systemd offers you to speed up the boot are useful to speed up session startup, and hence we'd want to make systemd useful for that job too. This is not a completely new idea though. For example on MacOS launchd is used in a similar way to start up both the system and the session.

That all said, along the way we'll fix a couple of other things too. For example, we want to make sure that ordering system services into cgroups is complemented by doing so for user services. Also we want to fix multi-seat support for good.

For more details see [this google doc](#).

**LinuxFr.org : In your FOSDEM interview you said: "I recommend developers to try to hack with only Linux in mind and experience the freedom and the opportunities this offers you". Do you understand the uproar of BSDs developers when they see this recommendation ?**

**Lennart :** Sure I understand that. It's not particularly surprising if they don't like recommendations like that, is it?

**LinuxFr.org : Systemd use a lot of Linux only technologies (cgroups, udev, fanotify, timerfd, signalfd, etc). Do you really think the Linux API has been taking the role of the POSIX API and the other systems are irrelevant ?**

**Lennart :** Yes, I don't think BSD is really too relevant anymore, and I think that this implied requirement for compatibility with those systems when somebody hacks software for the free desktop or ecosystem is a burden, and holds us back for little benefit.

I am pretty sure those other systems are not irrelevant for everybody, after all there are people hacking on them. I just don't think it's really in our interest to let us being held back by them if we want to make sure Linux enters the mainstream all across the board (and not just on servers and mobile phones, and not in reduced ways like Android). They are irrelevant to get Free Software into everybody's hand, and I think that is and should be our goal.

But hey, that's just me saying this. I am sure people do Free Software for a number of reasons. I have mine, and others have others.

**LinuxFr.org : Perhaps Debian will be unable to switch to systemd due to the necessity to support the alternative Debian GNU/kFreeBSD system. What do you think about this situation ?**

**Lennart :** Well, it's a loss for them, I guess.

Debian kFreeBSD is a toy OS, people really shouldn't misunderstand that. It's fine if people do toy OSes, everybody loves toys after all. But if Debian thinks it should limit its development by the dreamy interest in toy OSes of some of its developers, then this is their problem. If Debian was my project I'd try to focus on making (or keeping) it professionally relevant, and just forget about kFreeBSD, but I am not a Debian Developer.

That said, I see no reason why it shouldn't be possible to ship some BSD init system on kFreeBSD, and systemd on Debian's Linux version. A package shipping both a systemd unit file and a SysV init script is a workable solution and systemd will handle that just fine.

I think the claim that the need to support kFreeBSD on Debian was a blocker for adopting systemd by default is simply not true.

**LinuxFr.org : After reading your posts at LWN and elsewhere I've seen that you're not afraid of heated debates and flamewars. Do you enjoy this as part of the FLOSS folklore or is it really annoying ?**

**Lennart :** Not really, no. Being able to deal with poisonous people is part of being able to push big changes through in the Free Software world. And actually I doubt I am really that good in it, since I probably should stop much quicker replying to flamewars than I actually do. Most likely one should say "We have to agree to disagree on this" and shut up much earlier when a thread you are involved in turns into a flamewar.

**LinuxFr.org : Why Linux desktop hasn't been adopted by the mainstream users ? Linus Torvalds seems to think it's mostly a social issue and not a technical one. Do you agree with him ?**

**Lennart :** I think we weren't innovative enough in the interface, and we didn't have a convincing message and clear platform. If you accept MacOS as benchmark for user interfaces, then we weren't really matching it, at best copying it. I think this is changing now, with GNOME 3 which is a big step forward as an interface for Linux and for the first time is something that has been strictly designed under UI design guidelines.

So we now have a better interface, leaves the message and the clear platform. Linux is still too fragmented, and a developer targeting Linux will have to choose from a variety of APIs, a bazaar of somewhat matching but mostly just chaotic choices that will work on some systems but not on others. I think it would be in our greatest interest to streamline the platform top to bottom, and thus have a clear message what the Linux OS is. And of course, I believe my work in cleaning up the lower levels of our userspace stack is helping to work in that direction.

Getting a clear message out what Linux is supposed to be is definitely a social issue, but to make that happen the Linux platform needs to be streamlined first, and that's a technical task, and not done yet.

**LinuxFr.org : After all your projects are you proud of having changed the landscape of free software desktops ? Was it your motivation at the beginning ?**

**Lennart :** Sure I am proud. ;-)

Well, I simply like to hack, and I find closed software generally unpractical, so I do hope to play a small role in creating a competitive Free OS that can rival the existing ones in all and not just a few areas. And I am sure we'll get there.