INIT SYSTEMS

A brief history

Siddharth Tewari

- 2nd Year @ **PESU-ECC**
- Backend @ **Prevale**[upcoming]
- Mentor Team Co-lead @ **Hackerspace-ECC**
- Core team member @ **GDSC-ECC**
- Member @ **ACM-PESU**

So what is an init?

What is PID 1?

What are daemons?

init

It is the first process that starts when you boot in. It is the direct and indirect ancestor of all processes on your system

PID 1

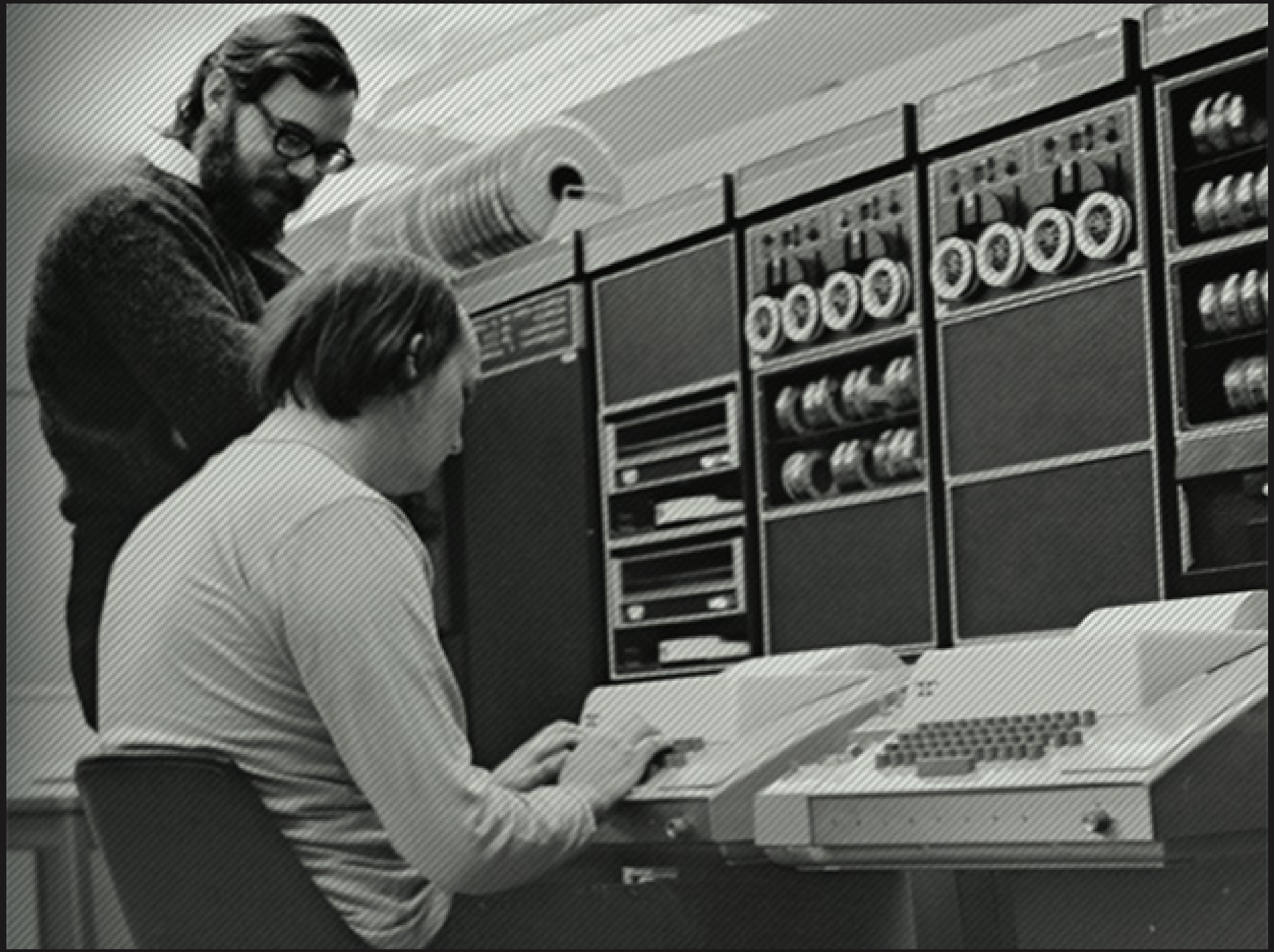
Process identifier assigned to the init daemon {launched via /sbin/init}

Is there a PID 0?

Daemons

Utility programs that run silently in the background to take care of and monitor subsystems

Where it all started



Ken Thompson and the late Dennis Ritchie at Bell labs in the 70s

Research Unix-style/BSD-style

- Essentially was an initialisation shell script
- no runlevels
- /etc/rc file determines what programs are run by init

- The advantage: simple and easy to edit manually.

- The issues: new software added to the system may require changes to existing files that risk producing an unbootable system

SysV init

- UNIX system III introduced a new style of system startup configuration
- It was modified and was introduced into UNIX system V

Modern day init

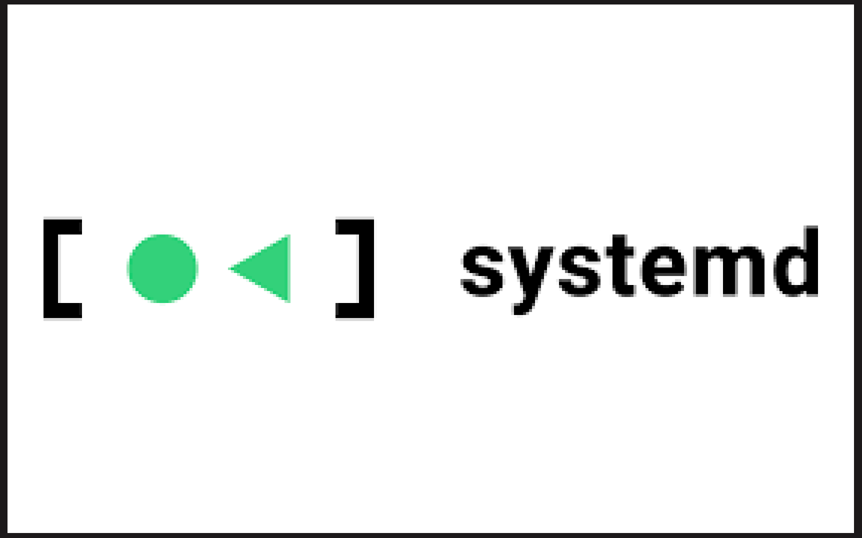
- Major drawback of init is that it was serial, waiting for each task to finish loading before moving on.
- When startup processes end up I/O blocked, this can result in long delays

Upstart

- Upstart focused on backward compatibility with SysV init so it could run unmodified SysV init scripts

- Widely adopted by major distros like fedora, ubuntu and some RHEL releases.
- Event-driven model responds to events asynchronously as they are generated

The rise of System-d

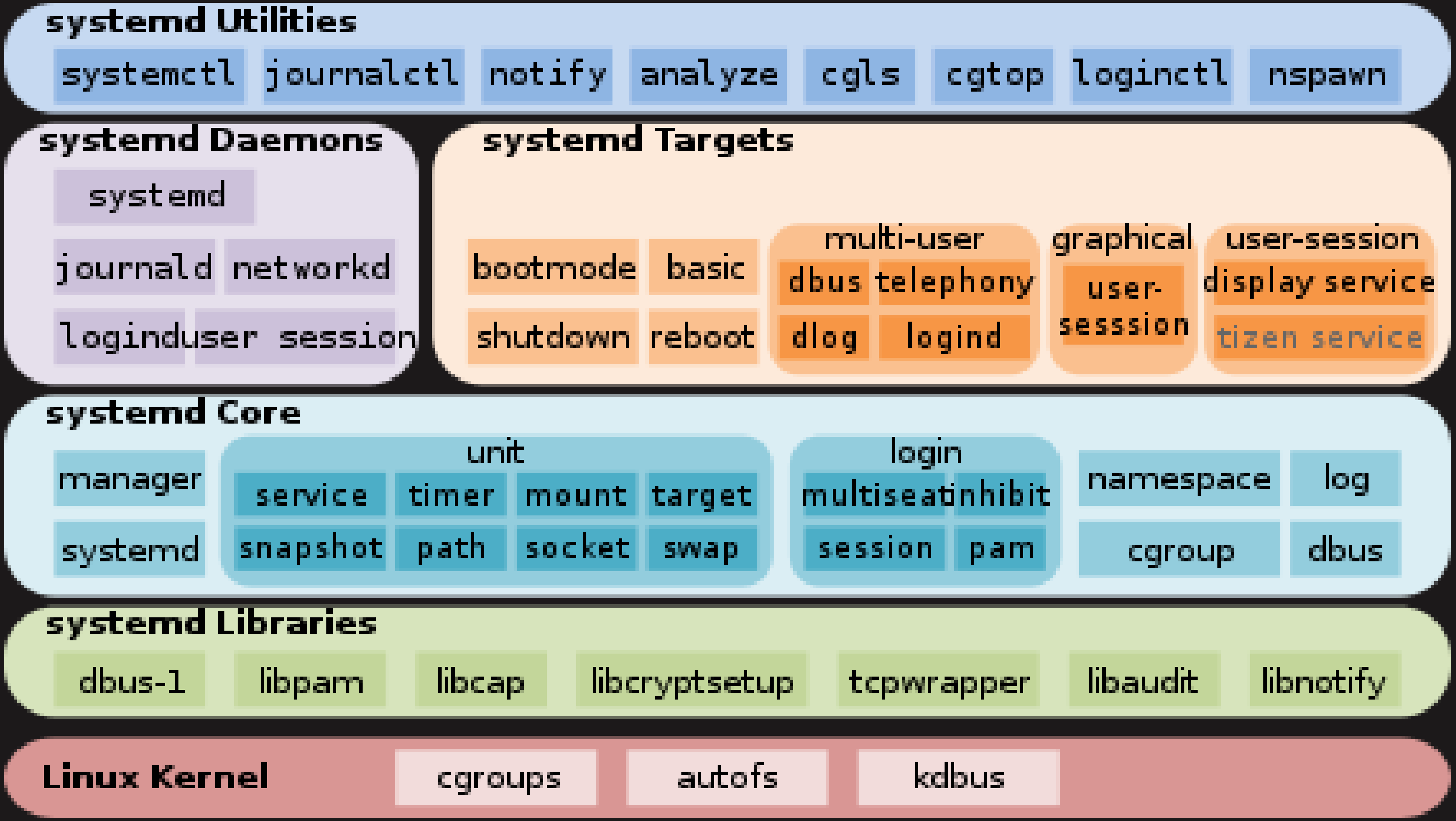


systemd
I'm going to change Linux world
For the better, right?
systemd
For the better, right?

Some of the most controversial figures of the Linux community in the 2010s

Lennart Poettering

Kay Sievers



systemd Utilities

systemctl journalctl notify analyze cgls cgtop loginctl nspawn

systemd Daemons

systemd

journald networkd

logind user session

systemd Targets

bootmode basic

shutdown reboot

multi-user

dbus telephony

dlog logind

graphical

user-sesssion

user-session

display service

tizen service

systemd Core

manager

systemd

unit

service timer mount target

snapshot path socket swap

login

multiseat inhibit

session pam

namespace log

cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

Meant to replace Linux's conventional SysV init(idea originated in 2010). Blog post (Rethinking PID 1)

- Fedora adopted first, in 2010
- Reasoning was: better parallelism and process management.

- In 2012, Arch Linux made the switch too from SysV init to system-d
- Some thought that the criticism towards systemd was not based on actual shortcomings of the software, and was rather the disliking of Lennart from a part of the Linux community and the general hesitation for change.

System-d itself was accused of:

- Being bloated
- Suffering from feature creep
- Violating the UNIX philosophy
- "Trying to take away freedom of choice by alienating and marginalizing other parts of the OSS community by uniformizing the Linux ecosystem."
- Being Monolithic rather than composable
- Having a large surface of attack(There's been numerous system-d exploits)
- Binary log files(supposedly leads to a lot of log corruption)

Poettering was hated.
Like a lot. It got to the point
that he started receiving death
threats over systemd.

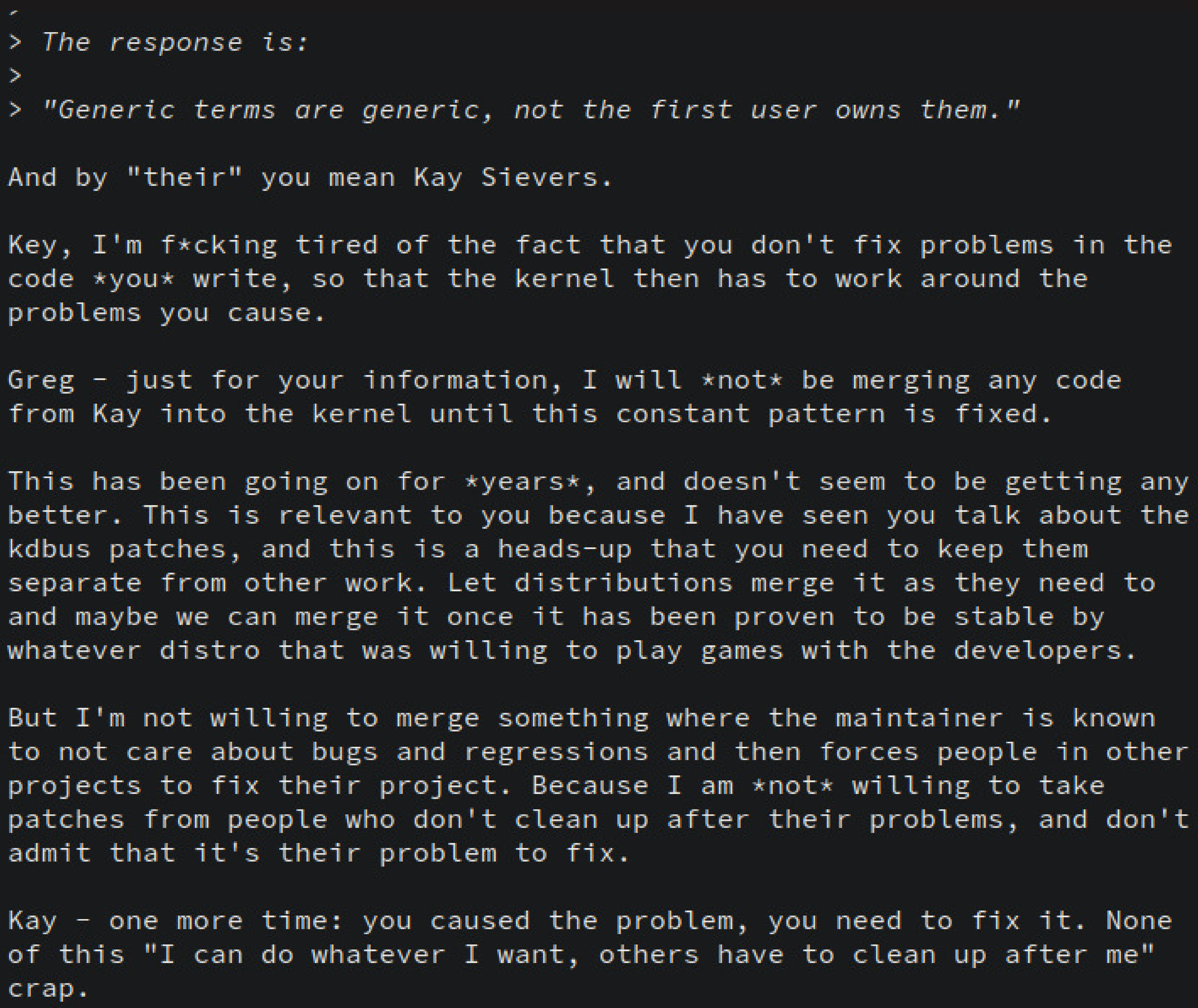
His ideas often brought accusations that he was working against the "unix philosophy",since he has advocated for breaking POSIX compatibility for speeding up Linux development

Linus Torvalds had issues with Kay Sieve previously

- Torvalds himself seemingly never had issues himself with system-d except a few nitpicks.
- He was moreso annoyed by the attitude of system-d maintainers towards bugs and users

"When it comes to systemd, they expect me to have a lot of colorful opinions, but no. I don't personally care about systemd, in fact my main computer and laptop use it. Now, I don't get along with some of the developers (referring to the Kay Sievers incident) and I think they are a little carefree when it comes to bugs and compatibility, but I'm not much in the camp of people who hate the idea of systemd either"

-Torvalds on systemd



> *The response is:*
>
> *"Generic terms are generic, not the first user owns them."*

And by "their" you mean Kay Sievers.

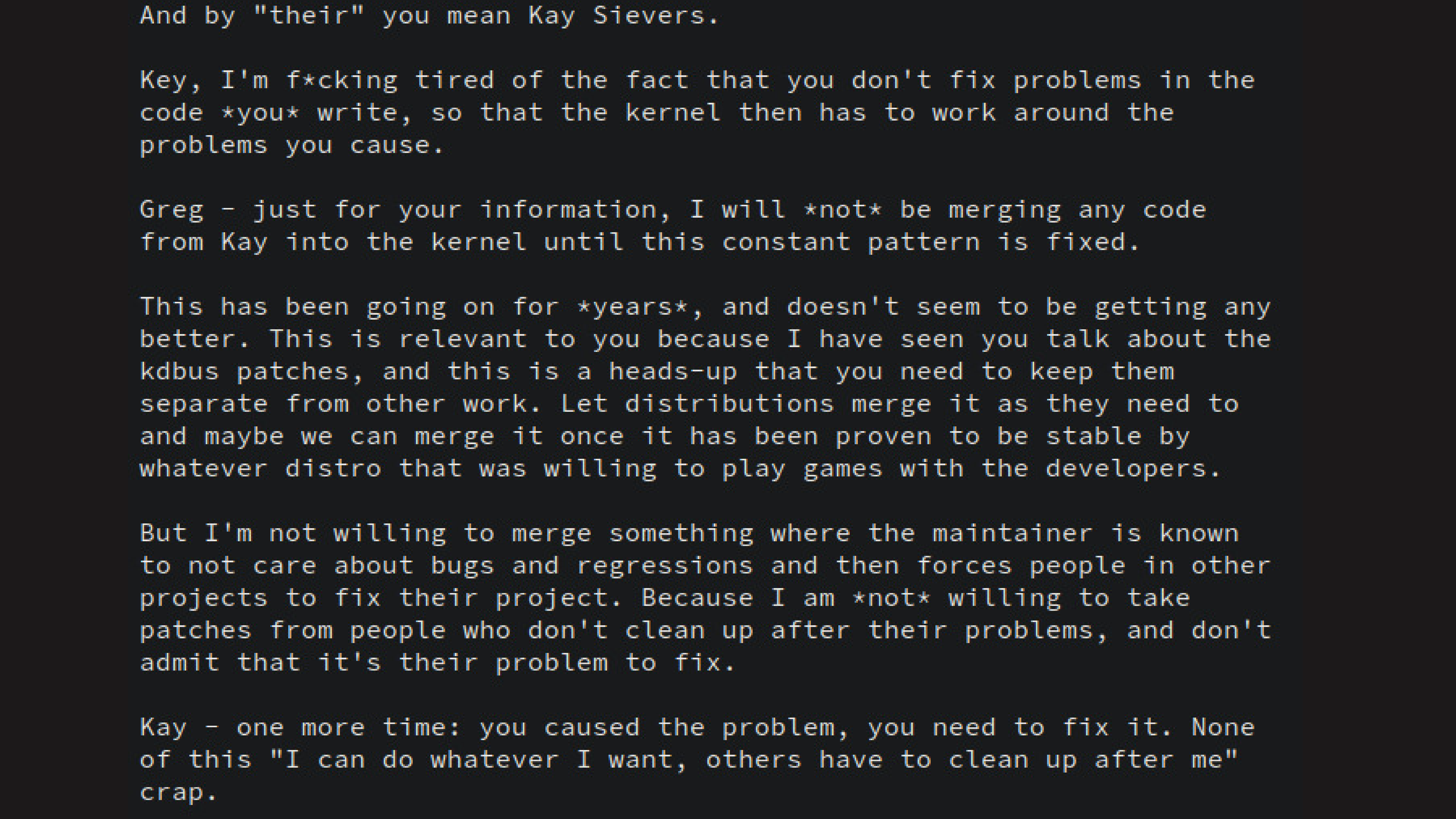
Key, I'm f*cking tired of the fact that you don't fix problems in the code *you* write, so that the kernel then has to work around the problems you cause.

Greg - just for your information, I will *not* be merging any code from Kay into the kernel until this constant pattern is fixed.

This has been going on for *years*, and doesn't seem to be getting any better. This is relevant to you because I have seen you talk about the kdbus patches, and this is a heads-up that you need to keep them separate from other work. Let distributions merge it as they need to and maybe we can merge it once it has been proven to be stable by whatever distro that was willing to play games with the developers.

But I'm not willing to merge something where the maintainer is known to not care about bugs and regressions and then forces people in other projects to fix their project. Because I am *not* willing to take patches from people who don't clean up after their problems, and don't admit that it's their problem to fix.

Kay - one more time: you caused the problem, you need to fix it. None of this "I can do whatever I want, others have to clean up after me" crap.



And by "their" you mean Kay Sievers.

Key, I'm f*cking tired of the fact that you don't fix problems in the code *you* write, so that the kernel then has to work around the problems you cause.

Greg - just for your information, I will *not* be merging any code from Kay into the kernel until this constant pattern is fixed.

This has been going on for *years*, and doesn't seem to be getting any better. This is relevant to you because I have seen you talk about the kdbus patches, and this is a heads-up that you need to keep them separate from other work. Let distributions merge it as they need to and maybe we can merge it once it has been proven to be stable by whatever distro that was willing to play games with the developers.

But I'm not willing to merge something where the maintainer is known to not care about bugs and regressions and then forces people in other projects to fix their project. Because I am *not* willing to take patches from people who don't clean up after their problems, and don't admit that it's their problem to fix.

Kay - one more time: you caused the problem, you need to fix it. None of this "I can do whatever I want, others have to clean up after me" crap.

Debian debate

- 2014: The vote is 4:4 between system-d and upstart in the debian technical committee
- Bdale Garbee, the chairman gives the casting vote in favour of system-d

Three months of discussions, two stalled votes, and one failed coup

Implications for canonical and ubuntu

Mark Shuttleworth, founder of canonical wasn't a big fan of system-d either, after canoncial had made upstart

"By contrast, those same outraged individuals have NIH'd [Not Invented Here] just about every important piece of the stack they can get their hands on ... most notably systemd, which is hugely invasive and hardly justified."

-Shuttleworth in late 2013

- They finally had to switch to system-d because it's "easiest" and probably "best".
- They get to drop a project and aren't fighting upstream.
- It also got them in line with other distributions (Red Hat, Fedora, etc) who are also moving to systemd.

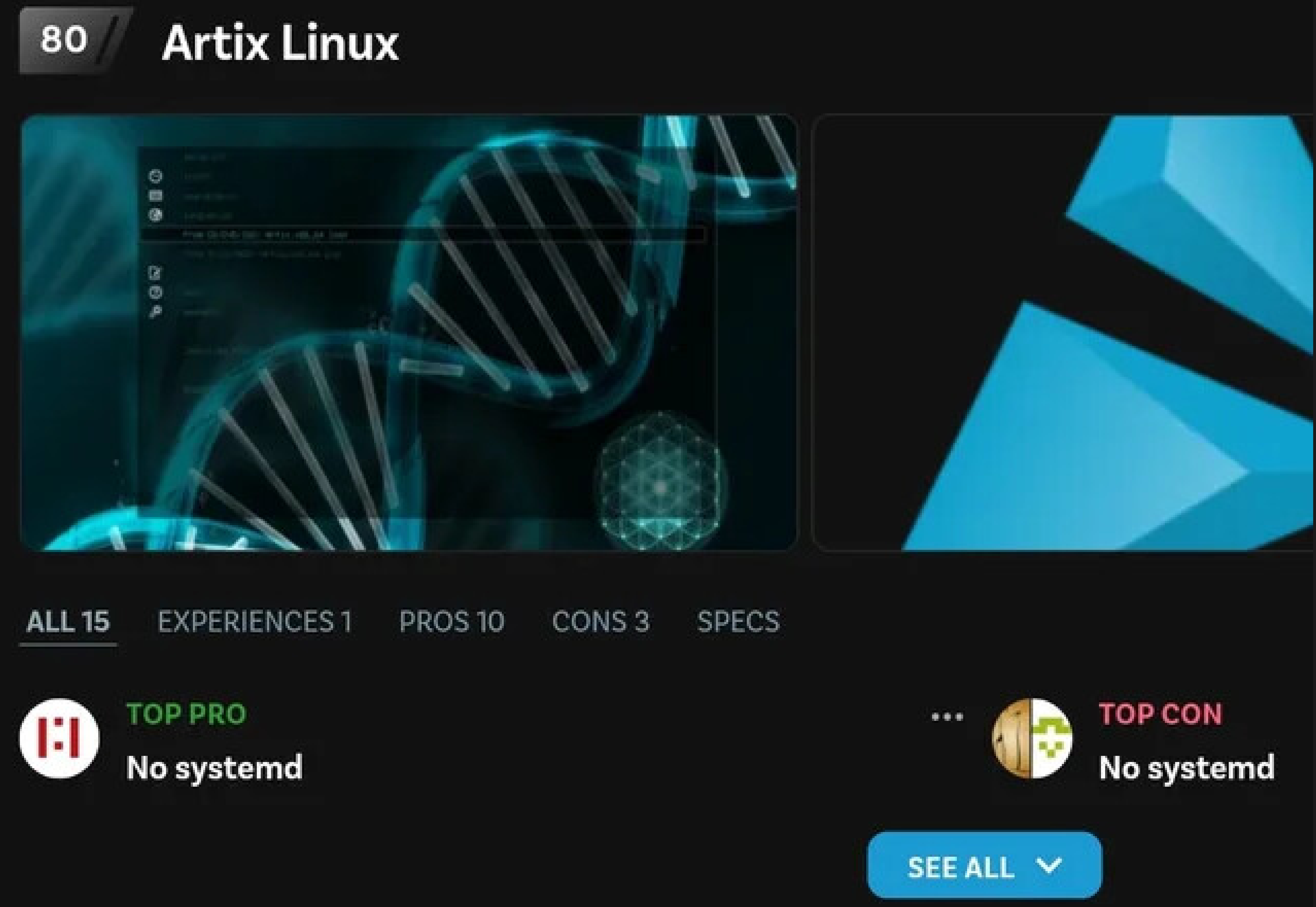
The case FOR systemd

"systemd is a great addition to the Linux server ecosystem. It's infinitely easier to write a systemd init script that is functionally correct, than it is to write the equivalent init script in shell."

-Comment on hackernews

- Its literally just used everywhere and in servers as well.
- To be able to work with systemd nowadays is a very basic skill

My experiences using runit



80

Artix Linux

TOP PRO

No systemd

TOP CON

No systemd

- Small, portable and modular codebase
- Initial release: 2004
- Drop-in replacement for SysV init

3 stages

- /etc/runit/1 - stage 1, system’s one-time initialization tasks
- /etc/runit/2 - stage 2, Normally runs runsvdir, should not return until the system is going to halt or reboot.
- /etc/runit/3 - stage 3, system’s shutdown tasks

Used shell scripts for starting services rather than systemd style unit scripts

Enable and add a service

```
$ ln -s ../../sv/service_name /run/runit/service
```

Disable a service

```
$ rm -f /run/runit/service/<service>
```

Point is:

Compared to systemd:

```
$ systemctl enable <service>
```

Runit syntax is a tab bit verbose

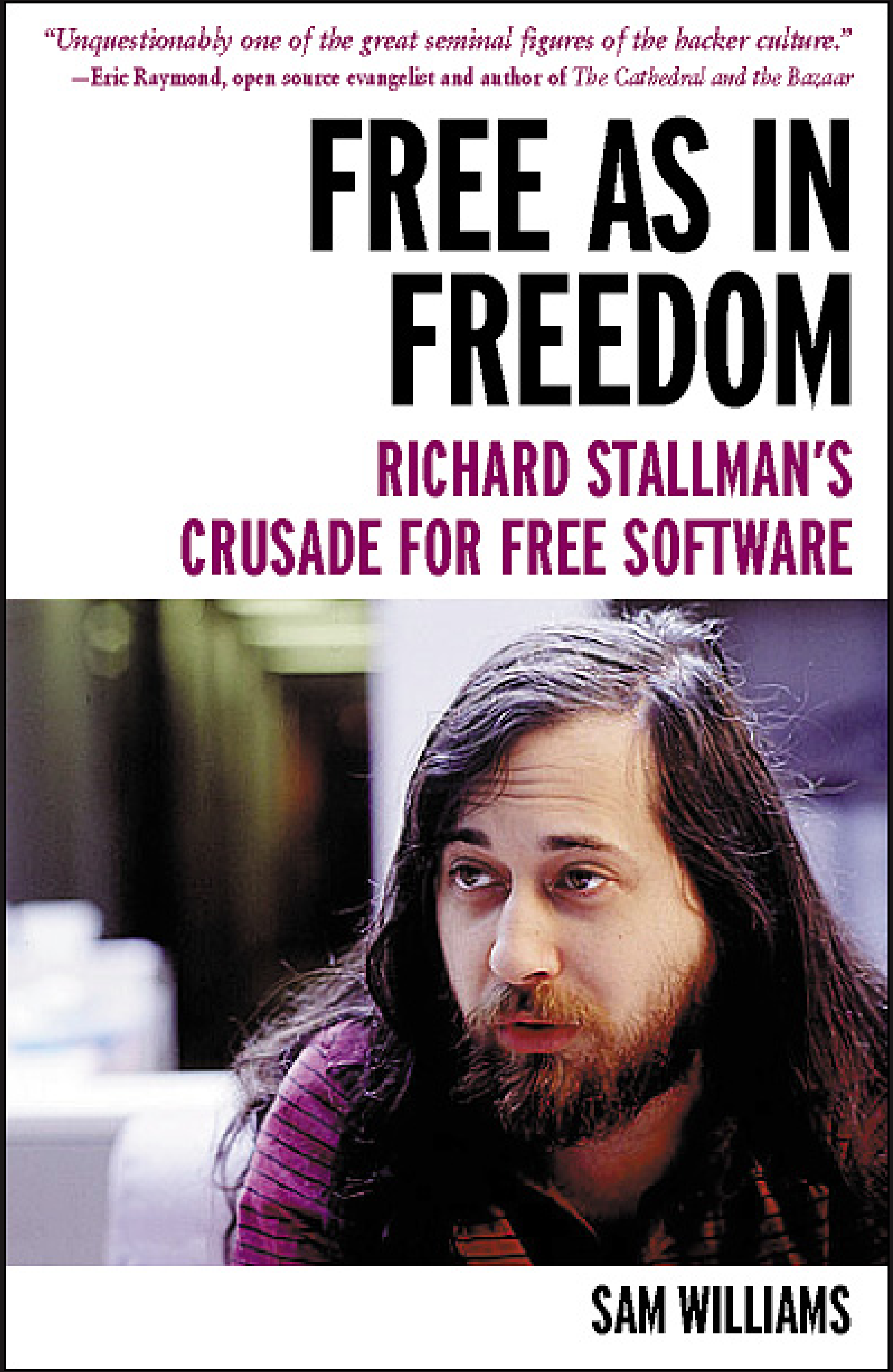
I often had to download runit scripts from the AUR or write my own for common things like docker or postgreSQL

In this regard, systemd offered far better ease of use and out of the box functionality.

And so I switched back

Regardless, having a choice in your init system is a bit important to some people

"Free as in freedom" and what not



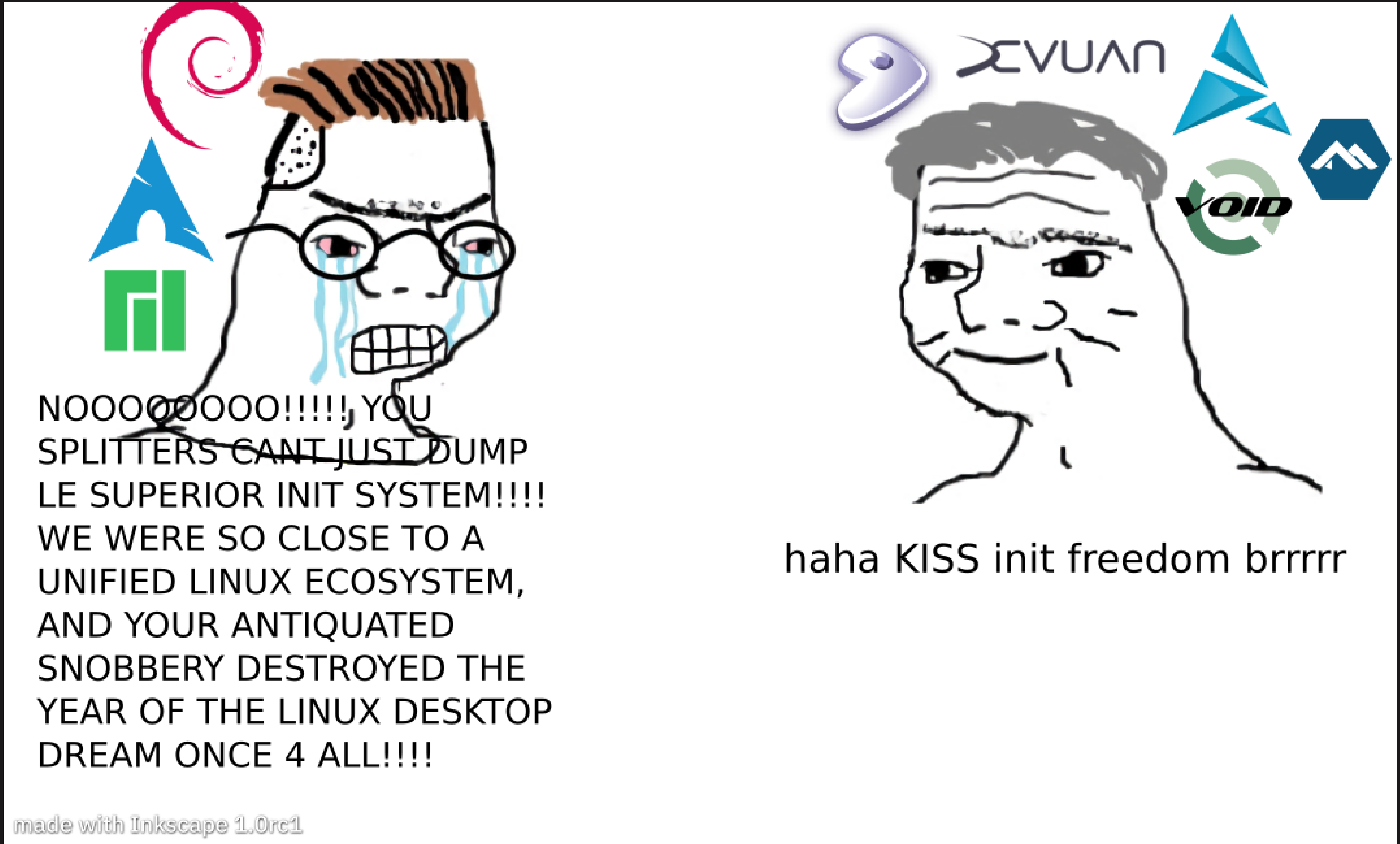
"*Unquestionably one of the great seminal figures of the hacker culture.*"
—Eric Raymond, open source evangelist and author of *The Cathedral and the Bazaar*

FREE AS IN FREEDOM

RICHARD STALLMAN'S CRUSADE FOR FREE SOFTWARE

SAM WILLIAMS

For those that didn't get the reference



NOOOOOOOO!!!!! YOU SPLITTERS CANT JUST DUMP LE SUPERIOR INIT SYSTEM!!!! WE WERE SO CLOSE TO A UNIFIED LINUX ECOSYSTEM, AND YOUR ANTIQUATED SNOBBERY DESTROYED THE YEAR OF THE LINUX DESKTOP DREAM ONCE 4 ALL!!!!
DEVUAN
VOID
haha KISS init freedom brrrrr
made with Inkscape 1.0rc1

Other modern init systems

Thank you!

That's all Folks!