

137 submitted 6 hours ago by Mcnst

46 comments share

all 46 comments sorted by: best

[-] meem1029 84 points 4 hours ago

Can someone tell me if I understand this situation correctly?

- Tmux uses the daemon system call to keep the process running in the background.
• This has historically kept processes running between user sessions.
• Systemd made the choice to change this behavior so that all such processes will be closed when the user logs off.
• This breaks much of the functionality of tmux.
• As a result of this, the systemd devs are now asking tmux to integrate a library to interact with systemd to fix this mistake.

If that's what's happening, I can certainly understand the hesitation on the part of tmux.

permalink embed

[-] thegenreeg 32 points 2 hours ago

As a result of this, the systemd devs are now asking tmux to integrate a library to interact with systemd to fix this mistake.

And this kind of approach is not something new...

permalink embed parent

[-] qftvfu 31 points 2 hours ago

Systemd broke background/daemon process behaviour, used by tmux, then asks tmux to fix/patch to accommodate the systemd change. I find this unacceptable and yet another example of scope creep from systemd.

permalink embed parent

[-] jking13 12 points an hour ago

I think scope creep is a bit of an understatement at this point. Of course given that Lennart Pottering has explicitly said he purposely wants things to not be portable (I thought Windows was supposed to be the platform that locks you in), so not surprising. Unless enough people who are paying money to RedHat tell them to stop (or just ditch it due to systemd), they'll keep trying.

permalink embed parent

[-] PM_ME_UR_OBSIDIAN 14 points an hour ago

Lennart Pottering has explicitly said he purposely wants things to not be portable

Source?

permalink embed parent

[-] thevilsharpie 8 points 49 minutes ago

https://lwn.net/Articles/430598/

Not having to care about portability has two big advantages: we can make maximum use of what the modern Linux kernel offers these days without headaches -- Linux is one of the most powerful kernels in existence, but many of its features have not been used by the previous solutions. And secondly, it greatly simplifies our code and makes it shorter: since we never need to abstract OS interfaces the amount of glue code is minimal, and hence what we gain is a smaller chance to create bugs, a smaller chance of confusing the reader of the code (hence better maintainability) and a smaller footprint.

permalink embed parent

[-] thebigslide 2 points 20 minutes ago

I tend to agree. Systemd does a lot of cool things well, but it simply isn't appropriate for many roles.

permalink embed parent

[-] aaron552 -3 points an hour ago

How is it scope creep to kill user processes on log out? That user-started daemons persist after logout by default should be considered a bug, not a feature

permalink embed parent

[-] _ak 15 points an hour ago

That user-started daemons persist after logout by default should be considered a bug, not a feature

Says who? This has been a Unix feature for the last 40 years or so, and is quite the fundamental aspect for a lot of essentials Unix tools and day-to-day operations.

permalink embed parent

[-] pjmpl 8 points 55 minutes ago

If you don't start them with nohup, they die. If I recall correctly.

permalink embed parent

[-] aaron552 5 points 47 minutes ago

is quite the fundamental aspect for a lot of essentials Unix tools and day-to-day operations.

Such as? (Don't say tmux or screen). Also, there's plenty of utilities (eg. ssh-agent) that run as daemons that should not persist after logout.

Note that I said user-started daemons. Most long-running daemons should not be running as ordinary users.

permalink embed parent

[-] HighRelevancy 1 point just now

DO YOU KNOW WHAT A DAEMON IS?

DO YOU EVEN

ARRHHGHGH

that user-started persistent processes happen to persist is a bug

WHAT THE HELL DUDE

WHAT EVEN WHAT ARE YOU SAYING

WHAT WERE YOU THINKING WHEN YOU PUSHED THAT INTO A KEYBOARD

permalink embed parent

[-] knaerwlmann 12 points 4 hours ago

Yep basically

permalink embed parent

[-] eigma 22 points 1 hour ago

But other daemons like ssh-agent also use daemon() and should be killed by systemd on logout.

So really the only fair statement we can make is: Linux does not have a precise enough process management API to support systemd's ultimate goal (which is laudable in isolation).

daemon() is not precise enough for this purpose, so I support the idea of having code in tmux to identify itself as a particular type of daemon. That said, I sympathize with tmux maintainers avoiding a dependency on dbus.

permalink embed parent

[-] Mcnst [S] 13 points 43 minutes ago

But other daemons like ssh-agent also use daemon() and should be killed by systemd on logout.

Source?

From what I can see, ssh-agent does not at all has any daemon() calls; in fact, to the contrary, it specifically installs a SIGHUP signal handler to terminate cleanly.

http://openbsd.su/src/usr.bin/ssh/ssh-agent.c#1385

1385 signal(SIGTERM, cleanup_handler);
1386 signal(SIGTERM, cleanup_handler);

permalink embed parent

[-] dvoget 3 points 38 minutes ago

If a lack of precision is truly the problem, the proposal to kill all descendants when the session exits seems overly broad. Sending SIGHUP as if the parent process exited would seem to achieve the terminate-unless-daemonized behavior that was lost with the dbus-ification of process management.

permalink embed parent

[+] lestofante comment score below threshold (1 child)

[-] icydocking 3 points 41 minutes ago

the daemon system call

There is no such thing. They reinvented the problem SIGHUP solves.

permalink embed parent

[-] morgan_lowtech 1 point 8 minutes ago

What's frustrating about this, and what's frustrated me about systemd for years, is that I generally agree with Lennart on a technical standpoint. I think what he and the other contributors are trying to do is valuable and needed; however, from a social standpoint it's just wrong.

Software is the interface of the machine and the meat and that means the messy, social, human component can't be ignored, even if the technical argument is superior.

Also, right now I feel like somehow this is an argument for Plan 9

permalink embed parent

[-] ksion 43 points 3 hours ago

The interesting part is why the default in systemd (i.e. of whether or not to kill user processes) was changed in the first place. As per this comment, it seems to be because of some lingering processes from Gnome login sessions. The commit which actually makes the change doesn't cite any other reasons besides the generic "much cleaner and provides better control". It is possible the actual reasons are sound and well-thought out, but the references provided by the person creating the issue fail to provide the sufficient rationale. It's not hard to understand the reluctance on the part of tmux's maintainers, especially given how much it's asked from them to accommodate for some other project's change which can very well be seen as frivolous.

permalink embed

[-] fubes2000 20 points 2 hours ago

I hope tmux stands their ground and refuses. I hate seeing open source projects being cowed by these pricks that seem to think that having one project in charge of everything is a good idea.

permalink embed parent

[+] bit- comment score below threshold (3 children)

[-] bit- 1 point 1 hour ago

But that is what one usually needs. Tmux and friends are special cases. Besides this already happens on desktop for the most part so why not go all the way and make sure user session is cleaned up. People hate change but this here is really non-issue.. Just some growing pains intensified by people who should run gentoo with openrc. .

permalink embed parent

[-] voice-of-hermes 0 points 12 minutes ago

I agree. If I want to stick a process in the background and keep it there, WTF are you doing trying to second guess me and kill it anyway?! That's some M\$-level operating-system-is-smarter-than-user bullshit, right there.

Gnome is broken? (Big surprise. /s) Let Gnome developers worry about that, not my whole freaking operating system.

permalink embed parent

[-] RubyPinch 12 points 3 hours ago

Reminds me of Windows in a way, user processes nuked on logout, services stay around

permalink embed

[-] Mcnst [S] 21 points 2 hours ago

Reminds me of Windows in a way, user processes nuked on logout, services stay around

Reminds me of UNIX in a way:

- User processes get sent an hup signal when the terminal disconnects, or if the parent process is the controlling process and performs an _exit system call. Default sigaction for SIGHUP (hup is short for hang-up) is to terminate the process.
• Daemons (including stuff like tmux) call daemon(3), which does some nifty things for you to ensure you become a service and stay around.

Déjà vu?

permalink embed parent

[-] RubyPinch 12 points 1 hour ago

Eh, daemon doesn't suddenly create service management in turn

I guess you could consider systemd's "shutdown every user process unless managed or otherwise specified" approach unixy, but it reminds me more of the central services management in Windows

Daemons seem more like a half-way point between processes and managed services

permalink embed parent

[-] lestofante 2 points 1 hour ago

Wait, daemon aren't systemd's services? Why can't daemon() create a managed systemd process?

permalink embed parent

[-] RubyPinch 8 points 56 minutes ago

Because it requires more information in general, e.g. dependencies, timeouts, potential actions for that service

Daemon is just like, "I want to stay open after term dies", nothing like "I also want to stick around when user logs out" or anything more specific

permalink embed parent

[-] lestofante 3 points 29 minutes ago

I still don't get it. Systemd can have a special type (or an existing type with settings tailored to specific case) of service with no respawn, dependency and all those thing that daemon() expect.. The call to daemon() then add this service. That mean that systems could also manage (and log?) User daemon, that make sense.

About saying more code.. Well, is better code in daemon() ONCE than DUPLICATION of code in many projects, no?

permalink embed parent

[-] Mcnst [S] 0 points 13 minutes ago

You're missing the point!

If everything directly depends on systemd, RedHat wins and makes \$\$\$\$\$\$\$\$. If fewer things directly depend on systemd, RedHat won't make as much money.

Guess which direction engineers on their payroll take?

Guess whose interests they ignore and who pays the price?

permalink embed parent

[-] Mcnst [S] 2 points 18 minutes ago

Because then it'll only require a single patch to libc, which daemon() library call is part of (libc/misc/daemon.c), instead of patching every single programme that needs to run it.

E.g., your suggestion would not make RedHat and their systemd achieve world domination as fast; plus I would imagine that the GNU libc maintainers won't be too thrilled about having a systemd in their list of things to deal with, either.

permalink embed parent

[-] lacosaes1 16 points 4 hours ago

I actually found the whole discussion very interesting. But I guess some people out there just want to talk shit about systemd every time they see a chance. To each their own I guess.

permalink embed

[-] IAMSnort 33 points 3 hours ago

Well how many outside dependencies do you want to maintain in your program?

permalink embed parent

[-] evotopid 6 points 42 minutes ago

Also: do you want to be part of forcing systemd on everyone?

permalink embed parent

[-] ledankmemes 3 points 35 minutes ago*

Noone complains when software is dependent on linux syscalls.

I mean sure some do, but if people care so much about code being dependent on systemd, then they should care more about it being dependent on linux.

permalink embed

[-] QtPlatypus 1 point 2 minutes ago

Not all Linuxes run systemd.

permalink embed parent

[-] skroll 1 point 1 minute ago

They are dependent on libc, not so much the syscalls. They may be implemented with syscalls, but it's completely different.

It would be more like "This software only works with musl libc"

permalink embed parent

[-] ECrownofFire 0 points 26 minutes ago

Could tmux not run as a different user and implement a way for user sessions to connect to it? Would solve this portably (hopefully) and remove the oddity of having a user daemon that sticks around.

permalink embed

[+] bubuopapa comment score below threshold (2 children)

search
this post was submitted on 30 May 2016
137 points (87% upvoted)
shortlink: https://redd.it/41n242
username password
remember me reset password login

discuss this ad on reddit
Submit a new link

programming
665,785 readers
878 users here now
/r/programming is a reddit for discussion and news about computer programming
Guidelines
• Please try to keep submissions on topic and of high quality.
• Just because it has a computer in it doesn't make it programming.
• Memes and image macros are not acceptable forms of content.
• If there is no code in your link, it probably doesn't belong here.
• App demos should include code and/or architecture discussion.
• Please follow proper reddit etiquette.

Info
• Do you have a question? Check out /r/programming, /r/cscaerquestions, or stackoverflow.
• Do you have something funny to share with fellow programmers? Please take it to /r/ProgrammerHumor/.
• For posting job listings, please visit /r/forhire or /r/Jobbit.
• Check out our faq. It could use some updating.
• If you're an all-star hacker (or even just beginning), why not join the discussion at /r/redditdev and steal our reddit code!

Related reddit
/r/technology
/r/learnprogramming
/r/askprogramming
/r/coding
/r/compsci
/r/dailyprogrammer
/r/netsec
/r/webdev
/r/web_design
/r/gamedev
/r/cscaerquestions
/r/reverseengineering
/r/startups
/r/techsupport
Specific languages
created by spez a community for 10 years

discuss this ad on reddit

MODERATORS
ketrlnis
spez
Poromenos
tryx
dons
masta
kylev
chromakode
a_redditor
about moderation team

discussions in /r/programming
1361 points - 209 comments
Github removes Streaks for good

about help apps & tools <3
blog about source code advertise jobs site rules FAQ wiki redditquiette transparency contact us
Reddit for iPhone Reddit for Android mobile website buttons
reddit gold redditgifts