systemd: Heresy and Hearsay

Tom Ryder
tom@sanctum.geek.nz
https://sanctum.geek.nz/

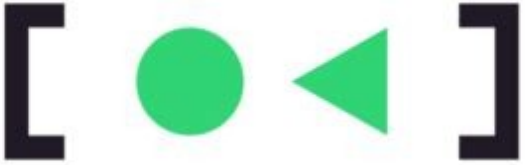
Holy wars

- Hackerdom (free software, “open source”…) has always had its holy wars.
- They are normally fairly tongue-in-cheek, and *relatively* good-natured debates.

  - emacs vs. vi vs. Nano
  - GNOME vs. KDE
  - Debian vs. Red Hat
  - GNU/Linux vs. BSD
  - Rust vs. Go
  - OOP vs. functional programming
  - Intel vs. AMD
  - Nvidia vs. AMD
  - IRC vs. XMPP
  - Perl vs. Python
  - Red Hat is good vs. Red Hat is evil
  - My computer vs. Your computer

`init(8)` of discord

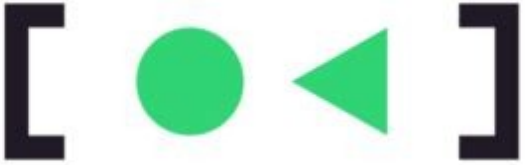
- The systemd debate **isn’t like that**.
- It gets *mean*.
- People take it *personally*.
- People are *heavily* invested in their opinion.
- Not to mention *dogmatic*…



What is systemd? 1/2

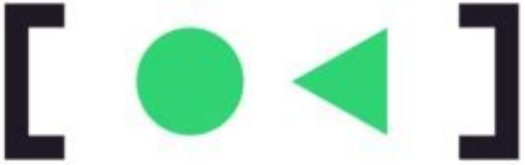
```
Welcome to Fedora 20 (Heisenbug)!

[  OK  ] Reached target Remote File Systems.
[  OK  ] Listening on Delayed Shutdown Socket.
[  OK  ] Listening on /dev/initctl Compatibility Named Pipe.
[  OK  ] Reached target Paths.
[  OK  ] Reached target Encrypted Volumes.
[  OK  ] Listening on Journal Socket.
         Mounting Huge Pages File System...
         Mounting POSIX Message Queue File System...
         Mounting Debug File System...
         Starting Journal Service...
[  OK  ] Started Journal Service.
         Mounting Configuration File System...
         Mounting FUSE Control File System...
[  OK  ] Created slice Root Slice.
```



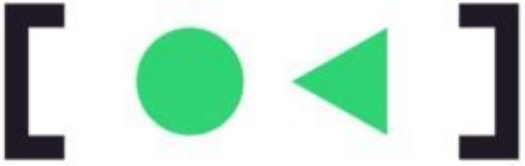
What is systemd? 2/2

- At its core: it's an `init(8)` implementation.
- It's the first user-space process started by your computer—**PID 1**.
- All other user-space processes descend from it.
- Other `init(8)`s include sysvinit, OpenRC…



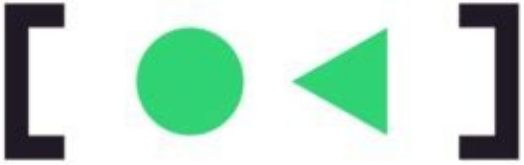
What does it do?

- *Lots* of stuff, for good or ill—we'll get to that…
- The most important part: it *supervises* processes, and lets the user manage them:
  - start/stop/reload your webserver (like `init.d`)
  - implement a scheduled backup (like `cron(8)`)
  - monitor resource usage of a group of processes
  - define the order in which services start



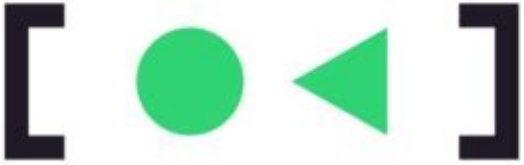
What’s good about it?—1/4

- It’s **free software**, under the GPLv2.
  - Just like the kernel itself.
- It’s **actively developed**, and not just by Lennart.
  - Give the other developers the credit they’re due!
- It’s **extensively documented**.
  - Including `man(1)` pages.



What’s good about it?—2/4

- It **speeds up boot** by defining how your services start.
  - Your webserver has to wait for your network stack to be ready…
  - …but systemd can start your system message bus…
  - …and your local disk mounts…
  - …and your TTYs…
  - …all in **parallel**.

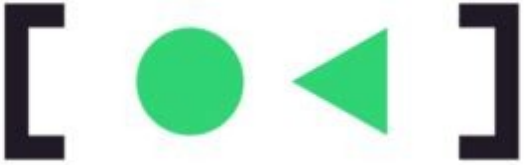


What’s good about it?—3/4

- It’s **built for the Linux kernel**.

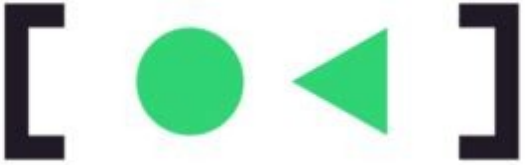
  (You may not consider that a good thing, but then, you’ve come to a LUG meet…)

  - ✓ **process control groups**
  - ✓ **network namespaces**
  - ✓ **mount namespaces**



What’s good about it?—4/4

- It **avoids writing shell script**.
  - Writing shell script for init processes is a pain.
  - It’s the same patterns with a few strings changed.
  - systemd makes this *declarative.*
  - You describe your service with a **config file**.

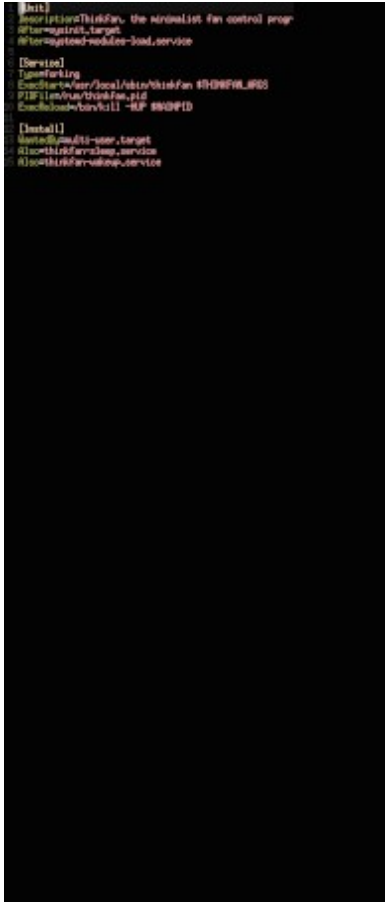


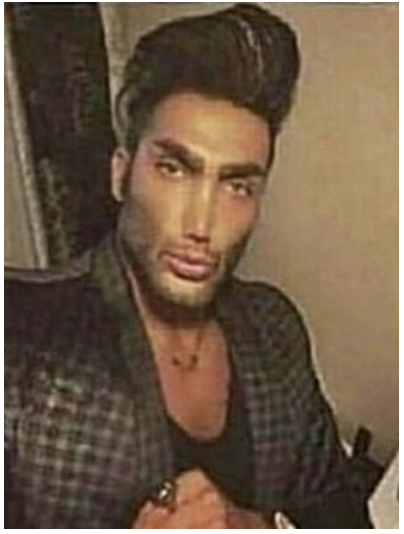
(and don’t forget, Tom *loves* shell script…)

Imperative services: the *how*

Average `init.d` script fan

Declarative services: the *what*





Average systemd unit enjoyer

So, what's the problem?—1/6

- systemd has little-to-no reverence for many *sacred cows* of standards and Unix culture:
  - Large suite of programs that do a lot
  - Threatens to replace programs that don't
  - Binary storage for logs (not text)
  - "What's POSIX, Precious?"

So, what’s the problem?—2/6

- systemd is **Unix heresy**.
  - But let’s not pretend it’s as bad as **macOS**…
- And where there’s heresy, there’s **dogma**.

So, what’s the problem?—3/6

- In itself, the heresy is **not the problem.**
- The problem is that systemd overhauls and changes *so much*, all at once…
- …that people flatly **refuse** to even *look* at it…
- …even though **there’s a lot to like**.

So, what's the problem?—4/6

- Not-hating systemd is a *mark of Cain* in some circles.
- It's just generally understood that *you simply don't use it*.
  - Doesn't matter why.
- Apologia for it is *inexcusable*.
  - Doesn't matter the reason.

"You *will* recant, Luther!"

So, what's the problem?—5/6

As a result, many critics of systemd I hear from don't know *anything* about it.

- They've never implemented a unit file for it.
- They've never *read* a unit file for it.
- "What do you mean there are `man(1)` pages?"
- systemd is just irredeemably bad.
- End of discussion.

So, what's the problem?—6/6

We've gone from **heresy** to **hearsay**:

>a problem happens with your computer

>someone steps in

>blames systemd

>blames you for using systemd

>refuses to elaborate

>leaves

Credit: ramkrsna

So, *bugger* all that.

“Hier stehe ich, und ich kann nicht anders.”

(Here I stand, and I can do no other.)

—Martin Luther, at the Diet of Worms
(likely apocryphal)

Let’s look at some *cool* stuff.

A simple service

- systemd **unit configuration files** look like INI files.
- They can be very short; files of six lines are typical:

```
[Unit]
Description=Examplenet network service

[Service]
ExecStart=/usr/local/bin/examplenetd --no-fork

[Install]
WantedBy=default.target
```

Timers—1/4

- Schedule a service to run every hour:

  `OnCalendar``=*-*-* *:00:00`

- Every minute:

  `OnCalendar=*-*-* *:*:00`

- At 7pm on the second Wednesday of every month except January, just in time for PLUG:

  `OnCalendar=Wed *-2..12-08..15 19:00:00`

Timers—2/4

- Combine multiple schedulers:

```
# ...after the first week of Jan, and
OnCalendar=Mon..Thu *-01-08..31 07:00:00
# ...every week from Feb through Nov, and
OnCalendar=Mon..Thu *-02..11-* 07:00:00
# ...up until the 20th of Dec.
OnCalendar=Mon..Thu *-12-01..20 07:00:00
```

Timers—3/4

- Five minutes after boot:

  OnBootSec=5m

- Thirty seconds after the last run *started*:

  OnUnitActiveSec=30s

- Two days after the last run *finished*:

  OnUnitInactiveSec=2d

Timers—4/4

- Add a random delay to avoid rushes of jobs every hour:

  `RandomizedDelaySec``=30m`

- You’d be surprised how much this can matter.
- Good manners for e.g. Certbot, too.

Dependencies

- Your custom server isn’t coming up on boot.
- It starts *too fast*, before all of the interfaces are bound.
- Tell it to wait for the network target:

```
Wants=network.target
After=network.target
```

Overrides

- My packaged rsync daemon listens only on one interface.
- Its `systemd` unit doesn't wait for the **network target**…
- …but *everything else* is correct.
- No need to edit or copy the whole unit file.
- Just fix that one thing with `systemctl edit`:

```
Wants=network.target
After=network.target
```

Private `/tmp`

- `/tmp` is usually a shared filesystem.
- Programs risk being able to trample on each other.
- Give the service a private `/tmp`:

  `PrivateTmp``=true`

- Done!
- Not just a subdir; it's `/tmp` *to the process*.

Socket activation

- I don’t need my Gemini service all the time.
- Have systemd listen on the Gemini TCP port.
- If a request comes in, buffer it and *then* start the Gemini service, until it’s ready for input.
- Lazy-loading services!
- **Bonus:** With socket passing, the service can be given a private network namespace…

Resource accounting—1/2

- How much bandwidth/CPU time is my service using?

```
IPAccounting=true
MemoryAccounting=true
CPUAccounting=true
```

Resource accounting—2/2

- Now `systemctl status` shows:

```
          IP: 210.6G in, 39.5G out
      Memory: 1009.6M
         CPU: 1month 5d 22h 35min 14.676s
```

- Can track *groups* of services this way as well, with service **slices**.

Constraints—1/3

- Your **backup service** needs to read your whole disk, as the `root` user.
- But it shouldn’t be able to *write* to the system…
- …*except* to its dedicated backup location.
- You can’t do this with traditional Unix permissions alone.
- You need `mount(8)` trickery.

Constraints—2/3

- systemd makes it easy:

```
ProtectHome=read-only
ProtectSystem=strict
ReadWritePaths=/var/local/backup
```

Constraints—3/3

- Limit system calls to a specific set:

```
SystemCallFilter=@system-service
```

- Limit IP traffic to localhost addresses:

```
IPAddressAllow=localhost
IPAddressDeny=any
```

- Limit memory usage:

```
MemoryMax=512M
```

Sola scriptura

- You might still end up disliking systemd.
  - No judgement here!
  - It has some *glaring* faults.
- But dislike it because you *looked at it*…
  - …or even better, *tried* it.
- Not just because the Pope said so…
  - …or some nerd you ran into on IRC.

Credit: John P Salvatore

whitequark @whitequark · Mar 27

How I Learned to Stop Worrying and Love systemd

13 12 187

Tom Ryder
@tejrnz

Replying to @whitequark

Ironically, it’s too big to be totally good or totally bad

12:27 PM · Mar 27, 2021 · Twitter for Android

https://twitter.com/tejrnz/status/1375786627230887937

Questions?

- systemd home page: https://www.freedesktop.org/wiki/Software/systemd/
- Common myths about systemd: http://0pointer.de/blog/projects/the-biggest-myths.html

Email: tom@sanctum.geek.nz
Website: https://sanctum.geek.nz/
Twitter: @tejrnz
Fediverse: @tejr@mastodon.sdf.org