systemd

"Init on Steroids"

Andreas Härpfer

(andreas.haerpfer@consol.de)

Consol-Akademie

v2.1, 28. April 2015

Agenda

Einführung: Historie, Features, ...

Basics: Dienste verwalten, Targets, User-Kommandos, ...

Demo: Einen eigenen systemd-Service bauen.

Logging in the 21st Century.

Advanced Features: Ein kleiner Ausblick, was sonst noch geht.

Linux-Init bisher

Sys V-Init

- Robuste, einfache Struktur, “well known”.
- Init-Skripte sind Shellskripte (oft viel doppelter Code).
- Sequentielle Ausführung (langsam).
- Reihenfolge über *S*- und *K*-Links in den Runleveln.
- Keine echten Dependencies.
- Kein Monitoring/Auto-Restart von Diensten.
- Nicht Hotplug-aware.

Andere Init-Systeme

- Upstart (Ubuntu)
- BSD-like
- ...

Linux-Init bisher (cont.)

Diverse Eigentümlichkeiten, z. B.: Wie restartet man Apache?

```
# apachectl restart
# /etc/init.d/apache2 restart
# service apache2 restart
# restart apache2
# service httpd restart
```

Was bietet *systemd*?

Features

- Monitoring/Automatische Service-Restarts.
- Dependencies zwischen Services.
- Parallelisierter/asynchroner System-Startup (schnell!).
- Eventbasierte Aktionen (*inetd* und mehr . . .).
- Deklarative Konfigurationsdateien statt Init-Skripte (einfacher, übersichtlicher).
- Funktionalität über Systemgrenzen hinweg (Remote Management, zentrales Logging) $\rightarrow$ “Think Cloud“

Anleihen u.a. bei

- *launchd* (Mac OS X)
- *SMF* (Solaris)

Adoption als Default-Init-System

Fedora	Mai 2011
openSUSE	September 2012
Arch Linux	Oktober 2012
CoreOS	Oktober 2013
Redhat Enterprise Linux	Juni 2014
Oracle Linux	Juli 2014
SUSE Linux Enterprise Server	Oktober 2014
Debian	geplant für Debian 8 “Jessie”
Ubuntu	geplant für 15.04
Gentoo	nicht als Default

https://en.wikipedia.org/wiki/Systemd#Adoption_and_reception

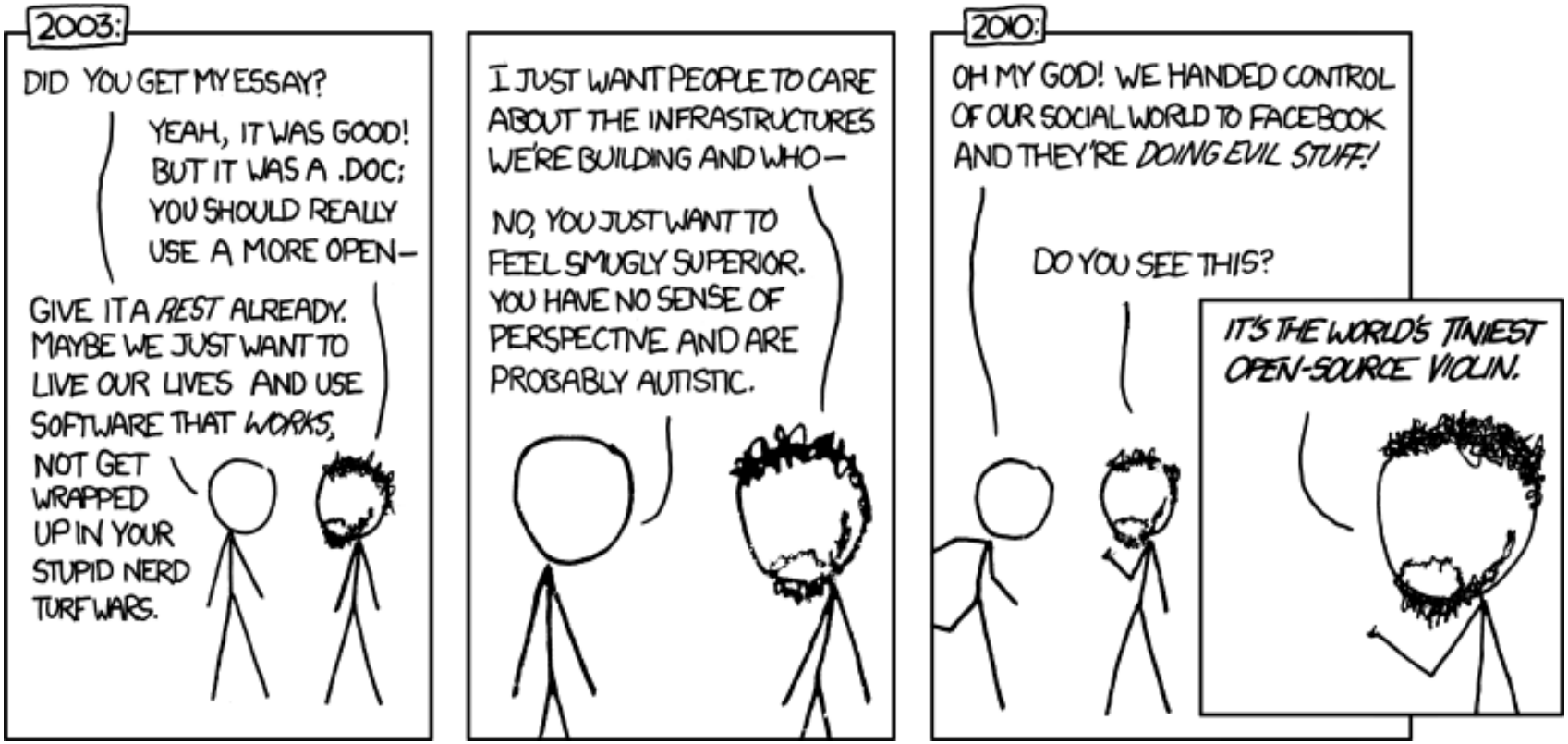
Kontroverse

- Bequem, modern, high-level, . . . aber . . .
- Verletzt KISS-Prinzip.
- Verwendet spezifische Features des Linux-Kernels
  $\rightarrow$ nicht auf anderes OS portierbar.
- Verleibt sich nach und nach andere Tools ein $\rightarrow$ Bloat.
  (D-Bus, *udev*, *inetd*, *cron*, . . .)
- Teilweise sehr kontrovers wahrgenommener Entwickler
  (Lennart Poettering [Redhat]).
- . . .

"It's a bit like if the makers of ovens decided that kitchens would be easier if the oven was also the fridge, kettle, freezer, dishwasher and sink."

http://www.linuxtoday.com/upload/is-systemd-as-bad-as. . .

"People who favour systemd generally use it to get shit done, not write blog posts about 'freedom of choice'". [1]

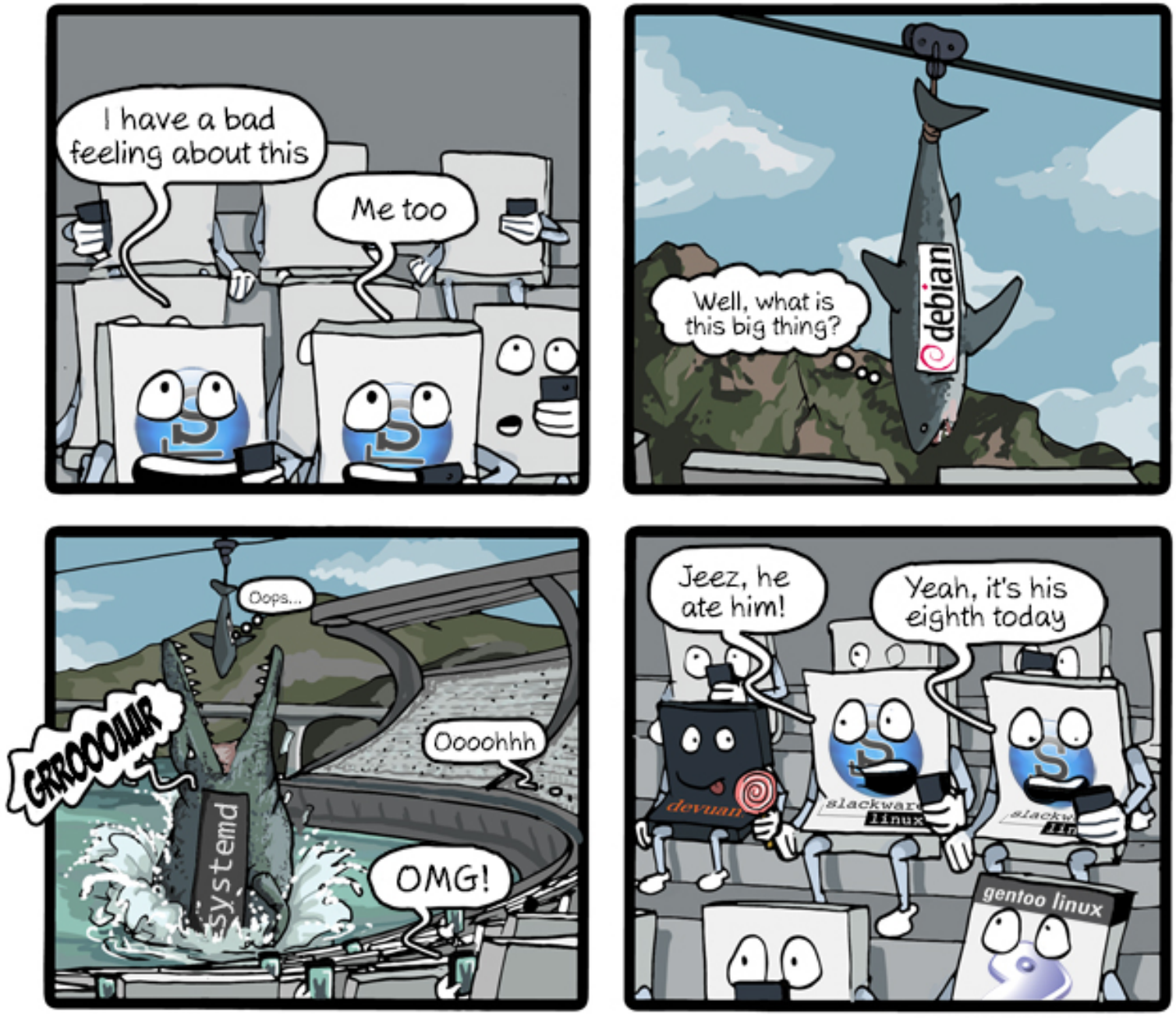


[1] https://news.ycombinator.com/item?id=7729075
https://xkcd.com/743/

Kontroverse (cont.)

"... In November 2014, Debian maintainers and Technical Committee members Joey Hess, Russ Allbery, Ian Jackson and systemd package maintainer Tollef Fog Heen resigned from their positions. All three justified their decision on the public Debian mailing list and in personal blogs with their exposure to extraordinary stress levels related to ongoing disputes on systemd integration within the Debian and open source community that rendered regular maintenance virtually impossible ..."

http://en.wikipedia.org/wiki/Systemd#History


I have a bad feeling about this
Me too
Well, what is this big thing?
debian
Oops...
GRROOOAAR
Oooohhh
systemd
OMG!
Jeez, he ate him!
Yeah, it's his eighth today
devuan
slackware linux
gentoo linux
CommitStrip.com

Komponenten/Nomenklatur

systemd

- systemd (PID=1)
- systemctl
- Unit-Files
- */etc/systemd/system.conf*

Unit-Files

- Pkgs: */lib/systemd/system*
- Lokal: */etc/systemd/system*
- “Drop-ins”: */etc/systemd/system/foobar.service.d/*.conf*

Unit-Typen

- target
- service
- socket
- mount/automount/swap
- path
- timer

Komponenten/Nomenklatur (cont.)

Logging

- systemd-journal
- journalctl
- */etc/systemd/journald.conf*

Weitere Tools

- loginctl
- hostnamectl
- busctl
- networkctl
- timedatectl
- localectl
- ...

Services – *systemctl* Cheatsheet

Übersicht

```
# systemctl
# systemctl list-units --type service
# systemctl list-unit-files --type service
```

Dienste starten, stoppen, etc. (nicht reboot-fest)

```
# systemctl start   foobar
# systemctl stop    foobar
# systemctl restart foobar
# systemctl reload  foobar
# systemctl status  foobar
```

Services – *systemctl* Cheatsheet (cont.)

Dienste reboot-fest aktivieren/deaktivieren:

```
# systemctl enable       foobar
# systemctl disable      foobar
# systemctl is-enabled   foobar
# systemctl mask|unmask  foobar
```

"Deep inspection" (Unit-Kontext)

```
# systemctl show foobar
```

Services – *systemctl* Cheatsheet (cont.)

Rekonfiguration (nach Installation neuer Unit-Files)

- Liest alle Unit-Files neu ein.
- Baut Dependency-Tree neu auf.
- Implizit bei *enable* und *disable*.

```
# systemctl daemon-reload
```

Boot-up performance

```
# systemd-analyze [time|blame|plot|...]
```

Targets

Was sind Targets?

- Definierte Laufzeitkonfigurationen ($\approx$ Runlevel/Milestone).
- „Ankerpunkte“ für Dependencies.

```
# systemctl list-units --type target
```

Target wechseln

```
# systemctl isolate some.target
```

Default Target auslesen/setzen

```
# systemctl get-default
# systemctl set-default some.target
```

Targets (cont.)

Runl.	Target Units	Description
0	runlevel0.target, poweroff.target	Shut down and power off
1	runlevel1.target, rescue.target	Rescue shell (single user)
2	runlevel2.target, multi-user.target	Non-graphical multi-user
3	runlevel3.target, multi-user.target	ditto.
4	runlevel4.target, multi-user.target	ditto.
5	runlevel5.target, graphical.target	Graphical multi-user
6	runlevel6.target, reboot.target	Shut down and reboot

Targets (cont.)

Maintenance Modes

Single-user: Inkl. lokaler Filesysteme und einiger wichtiger Services. *Kein Netzwerk!*

```
# systemctl rescue
```

Emergency-Mode: Nur *RO* Root-FS . . . falls auch Rescue-Mode nicht mehr funktioniert:

```
# systemctl emergency
```

Targets (cont.)

Power Management Kommandos

Old command	New command
`halt`	`systemctl halt`
`poweroff`	`systemctl poweroff`
`reboot`	`systemctl reboot`
`pm-suspend`	`systemctl suspend`
`pm-hibernate`	`systemctl hibernate`
`pm-suspend-hybrid`	`systemctl hybrid-sleep`

Die Kommandos `runlevel`, `telinit`, `shutdown`, `reboot`, ... sind noch vorhanden, sind aber nur Symlinks auf *systemctl*.

Abkupfern beim *sshd.service*

```
1  [Unit]
2  Description=OpenSSH server daemon
3  After=syslog.target network.target auditd.service
4
5  [Service]
6  EnvironmentFile=/etc/sysconfig/sshd
7  ExecStartPre=/usr/sbin/sshd-keygen
8  ExecStart=/usr/sbin/sshd -D $OPTIONS
9  ExecReload=/bin/kill -HUP $MAINPID
10 KillMode=process
11 Restart=on-failure
12 RestartSec=42s
13
14 [Install]
15 WantedBy=multi-user.target
```

(Das korrespondierende Init-Skript ist 160 Zeilen lang [Debian Stable]).

Ein eigener Service

hello.service

```
1  [Unit]
2  Description=The fantastic Hello webapp
3  After=network.target
4
5  [Service]
6  WorkingDirectory=/root/demos/hello-webapp
7  ExecStart=/usr/bin/gunicorn -b ':8000' --workers=2
       hello:app
8  KillMode=process
9  Restart=on-failure
10
11 [Install]
12 WantedBy=multi-user.target
```

Hilfreiche Manpages

systemd.unit(5): Globale Parameter für die Unit-Konfiguration

systemd.service(5): Parameter für Service-Unit

systemd.exec(5): Prozess-Environment (inkl. Settings für Logging, TTY, I/O- & CPU Scheduling, ...)

systemd.kill(5): Exit-Codes, Restart-Optionen

systemd.resource-control(5): Resoure Control Settings für die Prozesse des Service (später, advanced feature)

... und einige mehr

systemd-Logging

Features

- Binäres “Journal”.
- Strukturierte Logs inkl. Metadaten, Indizierung, . . .
- Logging auch wenn noch kein *rsyslog* läuft.
- Default: Weiterleitung an *rsyslog*, Journal nur im Memory.
- Separate Persistierung möglich.

```
# mkdir /var/log/journal
```

- Queries über Systemgrenzen hinweg.
- */etc/systemd/journald.conf*
- journald.conf(5)

systemd-Logging (cont.)

Kommandos

`# journalctl`	
`# journalctl -r`	Reverse
`# journalctl -p 4`	Prio $\geq$ Warning
`# journalctl -u sshd`	Unit-Filter
`# journalctl -n 1 -o verbose`	Metadaten
`# journalctl -o json-pretty`	als JSON
`# journalctl --since 12:00 --until 16:00`	Zeitfenster

Path-Monitoring

filemover.path

```
[Unit]
Description=Filemover watcher

[Path]
DirectoryNotEmpty=/var/tmp/in
#Unit=filemover.service

[Install]
WantedBy=multi-user.target
```

filemover.service

```
[Unit]
Description=Filemover service

[Service]
ExecStart=/root/demos/filemover/moveit.sh
Type=oneshot
```

cgroups und Resource-Management

- Linux Kernel-Feature um Resourcen von Prozessen oder Prozessgruppen zu limitieren.
- Eigene *cgroup*-Hierarchie unter: */sys/fs/cgroup/systemd*.
- Unterteilung in *Slices*, *Scopes*, *Services*.
- *systemd-cgls*: cgroups-Hierarchie anzeigen.
- *systemd-cgtop*: Top-artiger Output des Resourcenverbrauchs.
- systemd.resource-control(5)

Beispiel: I/O eines Service deckeln

mybackup.service

```
1  [Unit]
2  Description=mybackup
3
4  [Service]
5  BlockIOAccounting=1
6  BlockIOWriteBandwidth=/dev/sda 2M
7  ExecStart=/usr/bin/dd if=/dev/zero of=/backupfile
       bs=1024 count=1000000 oflag=direct
8  Restart=always
9
10 [Install]
11 WantedBy=multi-user.target
```

Referenzen

- RHEL 7 System Administrator's Guide (Kap. 6)
  https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/
- *systemd*-Manpages online:
  http://0pointer.de/public/systemd-man/
- *Das Init-System Systemd*, Lennart Poettering, Kay Sievers, Thorsten Leemhuis. Zweiteiliger Überblicksartikel auf *heise.de* mit einigen weiterführenden Links.
  http://www.heise.de/open/artikel/Das-Init-System-Systemd-Teil-1-1563259.html
  http://www.heise.de/open/artikel/Das-Init-System-Systemd-Teil-2-1563461.html
- Systemd-Projektseite auf *freedesktop.org*; viele weiterführende Links:
  http://www.freedesktop.org/wiki/Software/systemd/

Referenzen (cont.)

- *systemd for Administrators*, Serie von Artikeln im Blog von Lennart Poettering (mittlerweile 21 Teile!); viele Architektur-Details: http://0pointer.net/blog/archives.html
- *systemd in Debian*, Tollef Fog Heen, Michael Biebl, FOSDEM 2013 https://people.debian.org/~biebl/fosdem/debian-systemd.pdf

Fragen?