

# systemd

Michael „sECuRE“ Stapelberg

GPN12, 2012-06-08

powered by Lua<sup>A</sup>T<sub>E</sub>X

# Inhalt

- Motivation
- init
- systemd
- Flamewar?

# Motivation

Huch, mein Daemon läuft nicht mehr?

# Motivation

Ich habe Apache installiert, aber es startet nicht.

# Motivation

Ich habe `/usr/sbin/rpc.mountd --manage-gids`  
in meiner Prozessliste. Was ist das eigentlich?

# Motivation

Vielleicht 1x pro Monat drucke ich etwas. Warum läuft CUPS permanent?

# Motivation: Distro-spezifisches

- Starten mehrerer Instanzen (z.B. OpenVPN)
- (De)aktivieren von Diensten:  
`chkconfig, chmod -x, FOO_ENABLED...`
- Konfiguration von Parametern:  
`/etc/sysconfig/, /etc/default/, ...`
- Systemkonfiguration:  
Hostname, Locales, Module, `sysctl`, ...

# init

- PID 1, vom Kernel gestartet
- startet Scripts, Daemons, inkl. X11
- adoptiert elternlose Prozesse (fork), startet sie ggf. neu (z.B. getty)



# SysV init

- seit UNIX System V
- verschiedene „Runlevels“ geben an was gestartet wird:  
1 = single user mode, 2 = grafisches system, ...
- Reihenfolge (sequenziell) durch Nummern in `/etc/rc*.d`
- Shellscripts in `/etc/init.d/*` starten die daemons

# Probleme mit SysV init

- (leicht) verschiedene Scripts, je nach Distribution
- komplexe Scripts (125 Zeilen, 78 SLOC im Schnitt)
- jeder Daemon muss dasselbe implementieren:  
background, setsid, privilege dropping, PID files, log, ...
- daemons werden nicht überwacht (nur gestartet)
- langsamer (meist sequenzieller) Systemstart

systemd

# systemd

- kompatibel mit SysV init
- möglichst wenig (shell, unnötigen Code) starten
- Parallelisierung ( $\Rightarrow$  Geschwindigkeit!)
- **einfache** unit files ( $\cong$  initscript)
- implizite Abhängigkeiten, socket/dbus activation
- cgroups

# systemd Beispiel: thinkfan.service

[Unit]

Description=simple and lightweight fan control program

[Service]

ExecStart=/usr/sbin/thinkfan -q -n

[Install]

WantedBy=multi-user.target

# systemctl

```
michael ~ $ systemctl status thinkfan.service
thinkfan.service - simple and lightweight fan control pr
    Loaded: loaded (/lib/systemd/system/thinkfan.servi
    Active: active (running) since Sat, 13 Aug 2011 23
Main PID: 1794 (thinkfan)
    CGroup: name=systemd:/system/thinkfan.service
           1794 /usr/sbin/thinkfan -q -n

# start/stop/reload/restart, wie bei init.d
```

# Abhängigkeiten: socket activation

- socket activation für Socket-basierte Dienste wie syslog, D-Bus, CUPS, ...
- Beispiel: syslog nutzt /dev/log
- Daemons connecten darauf und schreiben Logmeldungen
- Idee: Socket **vor** Programmstart von syslog erzeugen
- Daemons können bereits connecten **und** schreiben, der Kernel buffert
- Sobald syslog läuft, kann er sofort die Daten wegschreiben

## Abhängigkeiten: socket activation (2)

- ⇒ Abhängigkeit ist implizit vorhanden. Wir können syslog und Daemons gleichzeitig starten
- Erst bei tatsächlicher Verbindung wird das zugehörige Programm gestartet
- ⇒ SSHd läuft erst, wenn man ihn tatsächlich benutzen will



# Abhängigkeiten

- Klassische Abhängigkeiten lassen sich via `After=` und `Before=` angeben
- Seltenst nötig (Beispiel: `ifup.service` setzt `After=local-fs.target`)
- Standardmäßig starten Services nach `basic.target`, welches low-level Zeug beinhaltet

# initscript-Inhalt

- Einige Scripts erledigen Aufräumarbeiten oder legen Verzeichnisse in `/var/run` an
- Stattdessen zentrale Config in `/etc/tmpfiles.d/*.conf`, systemd legt sie dann an/räumt auf
- Kernelmodule werden analog dazu in `/etc/modules-load.d/*` konfiguriert, sollten aber besser automatisch im Kernel geladen werden
- Kleinkram wie hostname setzen, Dateisysteme mounten/fscken sind direkt in C implementiert

# cgroups

- Saubere Trennung der Prozesse, eine cgroup pro service
- Man kann einen Service problemlos aufräumen, selbst wenn Teile davon crashen
- Resource-Limits (CPU, Speicher, ...) pro cgroup zuweisbar
- `systemd-cgls` listet die vorhandenen cgroups und deren Prozesse

# systemd-analyze

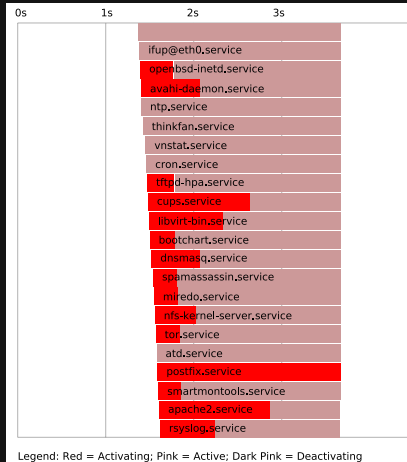
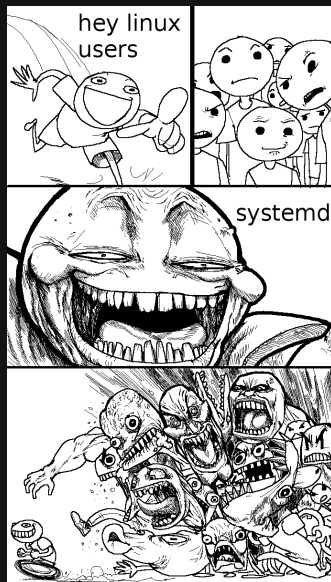


Figure: systemd-analyze plot > plot.svg

# Flamewar?



## Weitere Ressourcen

- <http://0pointer.de/blog/projects/systemd.html>
- <http://www.freedesktop.org/wiki/Software/systemd>
- <http://en.wikipedia.org/wiki/Systemd>
- <http://wiki.debian.org/systemd>
- <https://github.com/falconindy/systemd-arch-units>