



## Erste Erfahrungen mit systemd

---

**Stefan Bauer**

`stefan.bauer@cs.tu-chemnitz.de`

12. Oktober 2012



# Inhaltsverzeichnis

---

- 1 Motivation
- 2 Klassische Init-Systeme
- 3 systemd: Konzepte
- 4 systemd im Alltag
- 5 Neue Perspektiven
- 6 Fazit



## Aufgaben eines Init-Systems

- Starten und Stoppen der Hintergrunddienste
- Mounten der Dateisysteme
- Herstellen der Netzwerkverbindung
- Initialisierung der Hardware (Tastatur, Konsole, Mixer-Level)



## System V Init

- Kernel startet init-Prozess mit PID 1
- init startet alle zum Runlevel gehörenden Dienste
- ...streng sequentiell



## Beispiel

**Mount rootfs**  
**fsck Datenpart.**

**Mount rootfs**  
**fsck ifconfig**  
**sshd ntpdate kdm**

**ifconfig**  
**sshd**  
**ntpdate**  
**kdm**



# Probleme

- Explizites Festlegen der Reihenfolge
- Systemstart schwer paralellisierbar
- Statisches System
- Uneinheitliche Init-Skripte
- Kein sicheres Beenden von Diensten



# Neue Begriffe I

## Service

---

- Klassischer Hintergrunddienst
- Einmalige Ausführung

## Target

---

- Gruppierung von Services
- Ähnliches Konzept wie Runlevels
  - multi-user.target, graphical.target



## Neue Begriffe II

Noch vieles mehr (device, swap, mount, timer, ...)

### Unit

---

- Abstraktes Etwas
- Verwaltet durch systemd
- Konfiguriert in einer Datei



## Socket-Activation

- Ausnutzen blockierender IPC-Mechanismen zur Synchronisation
- Sockets werden von systemd angelegt und verwaltet (schnell)
- Wenn Service gestartet (langsam), Übergabe des Socket an den Service
- In der Zwischenzeit: Anstarten abhängiger Dienste



## Socket-Activation: Beispiele

### CUPS

- Socket: `/var/run/cups/cups.sock`
- CUPS wird erst gestartet, wenn etwas gedruckt werden soll

### sshd

- Socket: TCP Port 22
- sshd wird erst gestartet, wenn Verbindung eingeht
- sshd wird danach automatisch beendet



# Automount

- Spät benötigte Dateisysteme im Hintergrund einbinden
- Beim ersten Zugriff auf Dateisystem blockiert zugreifender Prozess



## Kommandos: Administration

### Aktivieren / deaktivieren eines Service

```
systemctl [enable|disable] NetworkManager.service
```

### Starten / stoppen eines Service

```
systemctl [start|stop] NetworkManager.service
```

### Wechsel des Runlevels

```
systemctl isolate graphical.target
```



---

# Kommandos: Analyse und Fehlersuche

## Systemstart analysieren

```
systemd-analyze plot > plot.svg
```

# Demo



# Konfiguration

- Symlinks in `/etc/systemd/system`
  - Welche Units sind aktiviert?
- Konfigurationsdateien in `/etc/systemd`
  - Laufzeitverhalten



## Control Groups I

- Fassen Prozesse in hierarchischen Gruppen zusammen
- (Ressourcenbeschränkungen für diese Gruppen)
- Sauberes und verlässliches Beenden von Diensten



## Control Groups II

**Control Groups der Units anzeigen**

---

`systemd-cgls`



# Perspektiven

- Starke Integration in den gesamten “Plumbing Layer”
  - Verschmelzung mit udev (Hardware-Hotplugging)
  - Übernahme von Logging-Funktionalität
  - Ablösen von Console-Kit
  - Verarbeiten von ACPI-Ereignissen
- Multi-Seat



# Journal I

- “Reichere” Log-Meldungen, Metadaten, Zeitstempel
- Zentralisiertes Logging (syslog, dmesg, Anwendungen)
- Kein Verlust von Meldungen beim Boot und Shutdown
- Log gegen Änderungen absichern



## Journal II

- In systemd integriertes Journal
- Binärformat
- Signiert



Fazit

---

## Wer braucht's?

- Anachronisten, die ihren Rechner ständig booten
- Server-Admins ([kernel.org](http://kernel.org))



Fazit

---

## Kritik

- Missachtung der Unix-Philosophie “One Task, one tool”
- Damit schlechte Austauschbarkeit
- Schlechter nachvollziehbar, schwerer zu debuggen
- Log im Binärformat

Vielen Dank für die  
Aufmerksamkeit!



---

## URLs

`http://www.freedesktop.org/wiki/Software/systemd`

`http://0pointer.de/blog/projects/systemd.html`

`http:`

`//0pointer.de/blog/projects/systemd-for-admins-1.html`

`https://docs.google.com/document/pub?id=`

`1IC9yOXj7j6cdLLxWEBAGRL6wl97tFxfgjLUEHIX3MSTs`