



Admin's Diary ...

Wie ich lernte, systemd zu lieben ;-)



Secure Linux
Administration
Conference

Berlin, 25.06.2015

Bjørn Bürger
bbu@pengutronix.de

motp @ {IRCnet, freenode, hackint, wunder-nett}

0183 3553 7110 3405 10B5 1CF6 4F4B B9AE 2074 ED8E
5B10 E8EB 86B7 2F61 65E6 21EA 41D4 4696 088D AB19



Was dieser Vortrag nicht sein soll



<http://cgit.freedesktop.org/systemd/systemd/tree/NEWS>



bbu:~\$ █





Bjørn Bürger

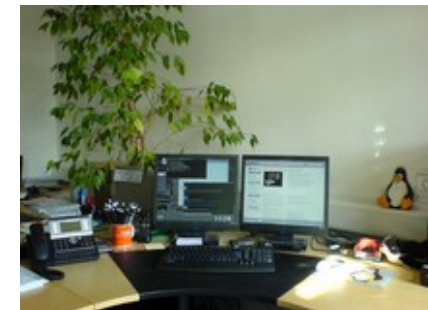
- 80er Jahre
 - Erster Computer-Kontakt
 - UniFlex (S-Plus South West Computer / MC68B09 / 2 MHz)
 - AMSDOS (Schneider CPC 6128 / Z80 / 4 MHz)
- 90er Jahre
 - Atari ST / PAK68 Linux m68k
 - PC SuSE 4.2 [...]
- TU Braunschweig
 - Linux User Group BS
 - Braunschweiger Linuxtage
 - Studenten-Netz
- Seit 2003
 - Embedded Linux / Entwicklung
 - Infrastruktur / Admin



Pengutronix

- Dienstleister
 - Embedded Linux Projekte
 - Free/Libre Open Source Software
 - > 2500 Beiträge im offiziellen Linux Kernel
- Seit 2001 in Hildesheim
- 20 Festangestellte

- Infrastruktur: Debian GNU/Linux
Zentrale Server-Infrastruktur
Remote-Labore
Desktop-Clients: PC





Früher™ war alles besser!

```
root@hobbes:~# apt-get install network-manager  
Paketlisten werden gelesen... Fertig  
Abhängigkeitsbaum wird aufgebaut.  
Statusinformationen werden eingelesen.... Fertig
```

**WARNUNG: Die folgenden essentiellen Pakete werden entfernt.
Dies sollte NICHT geschehen, außer Sie wissen genau, was Sie tun!
bjoerns-sanity**

Sie sind im Begriff, etwas potentiell Schädliches zu tun.
Zum Fortfahren geben Sie bitte »Ja, tue was ich sage!« ein

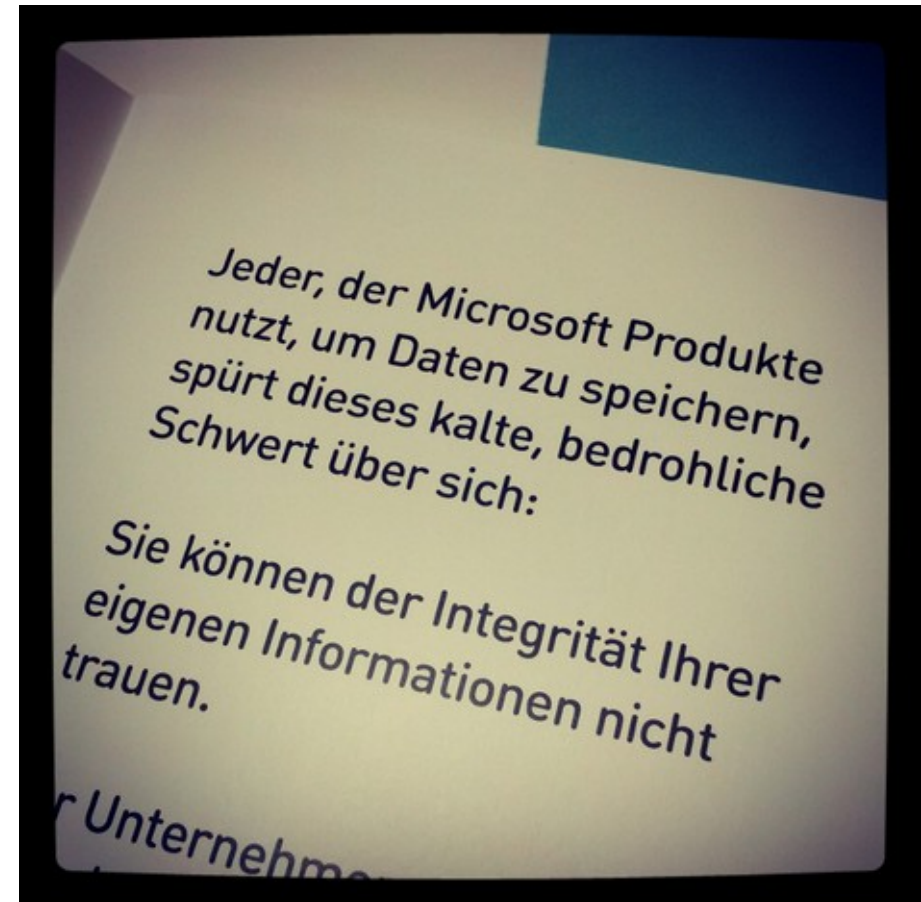
Die bestehenden Systeme sind komplex
und ständig kommt etwas „neues“ um die Ecke!

- Ablehnung!
- HabenWirDochAllesSchonGehabt!
- Beständig ist nur der Wandel
- Pengutronix Technikwochen
- "Mit systemd entsteht ein neuer Mechanismus,
ein Linux System hochzubringen" (2011)



Somebody is **WRONG** on the internet!

- Meinungen
- Wünsche
 - User
 - Admins
 - Entwickler
- Realität
- Ein Beispiel





User / Desktop

- Schneller Start
- Plug n' Play
- Modernes Multiseat Management
- Sicherheit
- Privatsphäre

- Voraussagbares Verhalten
- Gute Dokumentation
- Leicht auffindbare Logfiles

- Session Management
 - für das System
 - incl. Powermanagement Features, Netzwerk, etc.
 - für die User Session

- Desktop Virtualisierung
- Modernes Power Management



Admins

- Robuster Start
- Robuster Betrieb
 - Uptimes von mehr als 300 Tagen keine Seltenheit
 - Vereinheitlichung der Konfiguration verschiedener Distributionen
 - Einbindung in Configuration Management
- Container Management
 - Hohe Service-Dichte
- Gute Konzepte aus der Unix-Welt
 - Viele kleine, spezialisierte Tools
 - Scripting als „Kleber“
 - Reimplementierung in C für mehr Performance
 - "Everything is a file"
 - In-System Dokumentation
 - POSIX
- Schlechte Beispiele aus der Unix-Welt vermeiden
 - Kommerzielle Spezial-Tools, fehlende Standards



(Embedded-) Entwickler

- Schneller Start
- "Service Manager"
 - Restart eines "toten" Service
 - oder Reboot, wenn das reproduzierbar nicht klappt
 - Recovery-Systems im Fehlerfall
 - softwaregestützte Ausfallerkennung (per service watchdog)
 - hardwaregestützte Ausfallerkennung (hardware watchdog)
- Management der Systemressourcen
 - Services nur bei Bedarf starten
 - Services nur abhängig von irgendeinem Hardware-Aspekt starten
 - ssh server nur während der Entwicklungsphase
 - Debug Console nur, wenn ein Jumper gesteckt ist
 - Upgrade, wenn ein bestimmter USB-Stick gesteckt ist
 - Serielle Konsole, sobald ein bestimmter USB-Serial Adapter steckt
- "Trusted Boot"
- R/O Dateisystem
- Remote Update



Realität: sysvinit

- nur ein simpler "Service Launcher"
- Den Rest bastelt sich jeder selbst zusammen
- Serverpakete: eigene Management-Daemonen
- Keine globale, automatisierte Fehlerbehandlung
- Oft noch nichtmal sinnvolles Logging
- Jeder Admin hat seine eigenen "magischen" Scripte
- Spaghetti-Code in Init-Scripten
- Ausführbarer Code statt Config in /etc
- Keine saubere Trennung zwischen / und /usr

Folge:

- Ein hängendes Script stoppt alles
- "exit 0" Debugging
- /usr oft nicht Read-Only
- etckeeper
- sysgit



Realität: Embedded

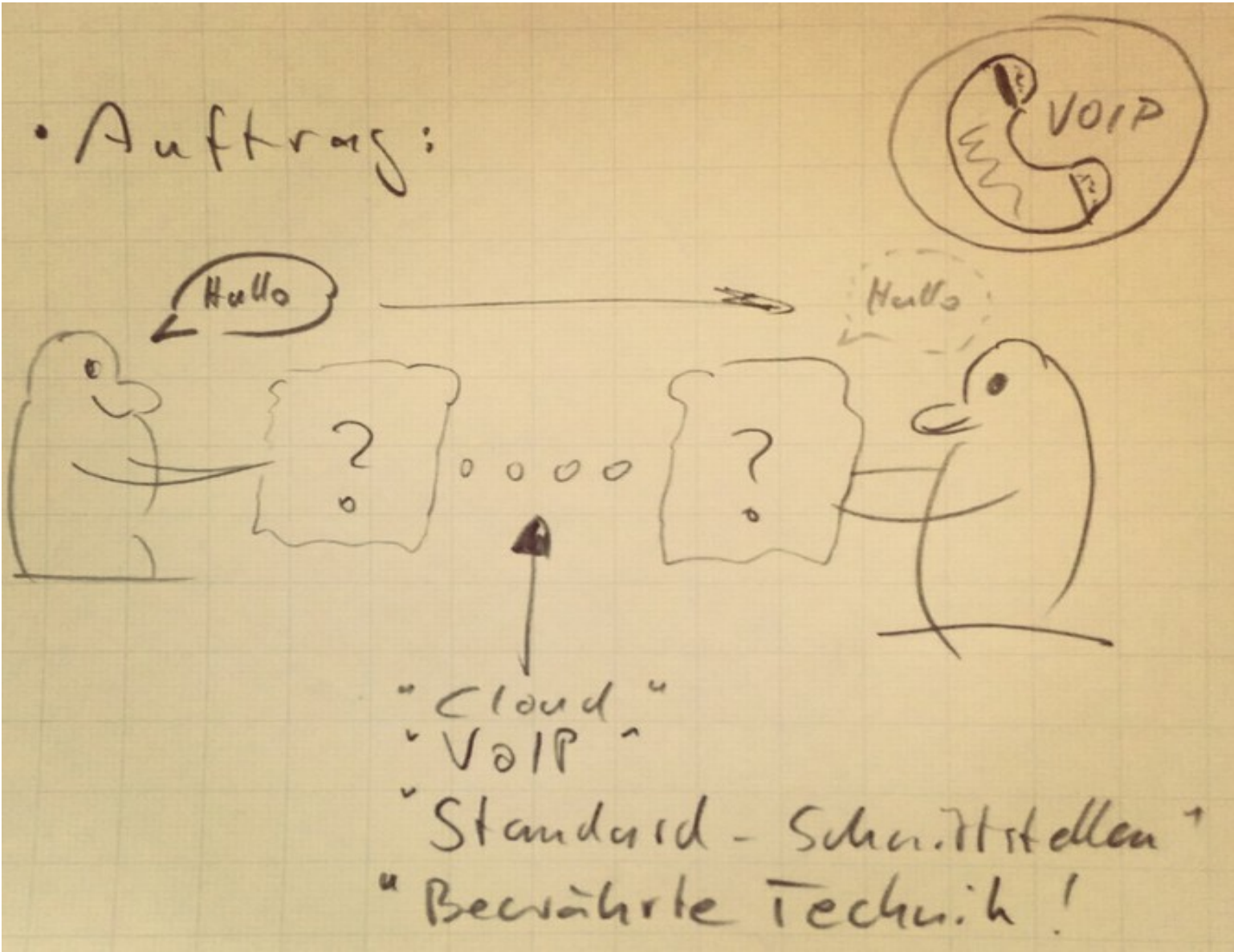
- Systeme ohne Monitor oder Tastatur
- Allenfalls minimale User-Interaktion

Folge:

- „Fire and forget“
- Manchmal nur ein riesiges Script
- Magie[tm]
- Der "superspezielle" Watchdog™ nach Art des Hauses
 - führt zu unnötigen Fehlern
 - fängt die entscheidenden Fehler nicht ab
- Dateisysteme sind in der Regel ReadOnly
 - /etc /var [...] per tmpfs (... komplett r/w)
 - Overlay Dateisysteme
 - Generator-Scriptmagie



Ein Beispiel





Bewährte Technik & Standard Schnittstellen



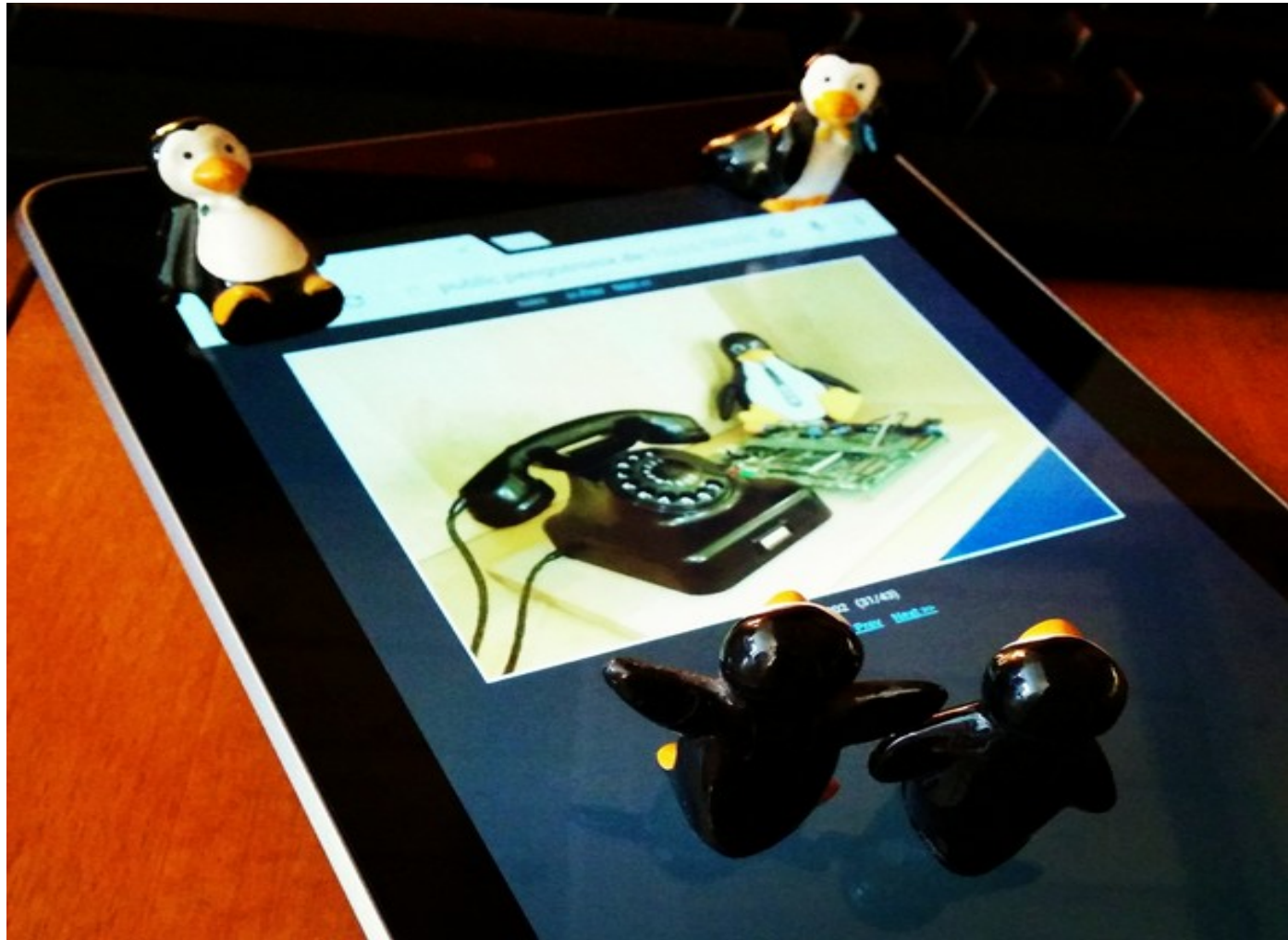


Modulare Netzwerk-Infrastruktur





Stand der Technik





Was lernen wir daraus?

- **Unix ist historisch für uns viel wert**
 - Inspiration
 - Konzepte
 - Vorlage
- **FLOSS darf und sollte seine Kernkomponenten nach Stand der Technik standardisieren.**
- **Es gibt kein GESETZ, das uns zwingt, auf immer und ewig mit Technik aus den 1960ern zu arbeiten.**
- **Es ist eine gute Idee, die Fehler der Vergangenheit nicht all zu oft zu wiederholen ...**





... und wenn doch ...
... dann wenigstens mit Spaß :-)



Die ganze Welt ist voll
von Sachen, und es ist
wirklich nötig, dass
jemand sie findet
Pippi Langstrumpf



systemd

- Ein System-, Service- und Session Manager für Linux
- Kompatibel mit SysV und LSB Init Scripts
- Wird inzwischen von allen neueren Linux Distros benutzt
- LFS: <http://linuxfromscratch.org/lfs/view/stable-systemd/>

- systemd nutzt die aktuelle Linux Infrastruktur
 - udev
 - inotify
 - dbus
 - cgroups
 - pam
 - Unix Resource Limits
 - Capabilities,
 - SELINUX, AppArmor, [...]

- Tracking von **Services und Sessions**, nicht nur von Prozessen
- Überwachung des Service während der gesamten Laufzeit
- Watchdog Kette von der Hardware bis zur Applikation



systemd - Für wen ist es gemacht?

- Server
 - Container Support
 - Lange Laufzeiten
 - Zentrales Management
- Embedded
 - „Headless“ Betrieb
 - „Unattended Upgrades“
 - Ressourcen!
- Desktop
 - Speed!
 - Desktops Session Management
 - Multiseat Support
- Deine Anwendung ist nicht dabei?
 - Send Patches!



systemd - Units

- Units sind die Building-Blocks des systemd Universums
- Deklarative Servicebeschreibung statt ausführbarer Dateien
- Sauberes Environment
- Socket-Aktivierung (aber auch inotify, dbus)
- Netzwerk-Services ohne Netzwerkzugang :o)
- Conditions

```
# /lib/systemd/system/cups.socket
[Unit]
Description=CUPS Printing Service Sockets

[Socket]
ListenStream=/var/run/cups/cups.sock
BindIPv6Only=ipv6-only
PassCredentials=true

[Install]
WantedBy=sockets.target
```

```
# /lib/systemd/system/cups.service
[Unit]
Description=CUPS Printing Service
Documentation=man:cupsd(8) man:cupsd.conf(5)
Requires=cups.socket

[Service]
ExecStart=/usr/sbin/cupsd -f
PrivateTmp=true

[Install]
Also=cups.socket cups.path
WantedBy=printer.target
```



Stateless Systems

- /usr -> Betriebssystem
- /etc -> Konfiguration (opt)
- /var -> State (opt)

- Ziel: Read-Only System im /usr (stateless)
- /etc ist nur noch minimale, lokale Konfiguration

```
.include /usr/lib/systemd/system/some.service
```

```
[Service]  
IOSchedulingClass=idle
```

- Alternativ: Droplets
/{run,etc,lib}/systemd/system/some.service.d/*.config

- Container-Instanzen können vom Basis-System "erben"
- YEAH :-D



systemd - timers

- potentieller cron und anacron Ersatz
- WakeSystem= -> Rechner per RTC aufwachen lassen
- OnUnitActiveSec= -> Zyklische Ausführung

```
events@calvin:~$ systemctl cat systemd-tmpfiles-clean.timer
```

```
[Unit]
```

```
Description=Daily Cleanup of Temporary Directories
```

```
Documentation=man:tmpfiles.d(5) man:systemd-tmpfiles(8)
```

```
[Timer]
```

```
OnBootSec=15min
```

```
OnUnitActiveSec=1d
```

```
events@calvin:~$ systemctl list-timers
```

```
NEXT                LEFT          LAST           \
Fr 2015-04-03 10:41:12 CEST 17h left Fr 2015-03-27 21:09:00 CET  \
```

```
PASSED          UNIT                ACTIVATES
5 days ago systemd-tmpfiles-clean.timer systemd-tmpfiles-clean.service
```



Nutzt die Sicherheits-Funktionen!

- Brot und Butter Technik ...
- Vieles geht auch mit sysvinit ... aber kaum jemand macht es!

- User, Group (su)
- RootDirectory= (chroot)
- Classic Unix Ressource Limits (ulimit)

- CapabilityBoundingSet= CAPABILITIES(7)

CapabilityBoundingSet=~CAP_SYS_PTRACE

- SELinuxContext=
- AppArmorProfile=
- SystemCallFilter=
- SystemCallArchitecture=
- RestrictAddressFamilies=
- CPUQuota=25%
- [...]



Nutzt Namespaces!

- ReadWriteDirectories=
- ReadOnlyDirectories=
- InaccessibleDirectories=
- PrivateTmp=Yes
- PrivateNetwork=Yes

```
[Service]
ExecStart=/lib/systemd/systemd-machined
BusName=org.freedesktop.machine1
CapabilityBoundingSet=CAP_KILL CAP_SYS_PTRACE [...]
WatchdogSec=1min
PrivateTmp=yes
PrivateDevices=yes
PrivateNetwork=yes
ProtectSystem=full
ProtectHome=yes
```

```
drwx----- 3 root root 4096 Mar  1 18:47
/var/tmp/systemd-private-ab..3bd13-systemd-machined.service-TSEHHU/
```

<http://www.freedesktop.org/software/systemd/man/systemd.directives.html>



systemd - Das Journal

- **syslog**
 - „Standard“ im RZ
 - steht im early-boot nicht zur Verfügung
 - war eigentlich immer „optional“
 - speichert oft unvollständig
 - speichert immer in der lokalen Zeitzone, notiert aber nirgends, welche Zeitzone verwendet wurde
 - ist nicht fälschungssicher
- **journal**
 - journald ist ... system log ... „neu gedacht“
 - deckt auch die early-boot Phase ab
 - rechnet Zeitangaben immer erst beim Anzeigen um
 - **Alles**, was ein Daemon rausschreibt, wird ins Log übernommen und passend indiziert
 - speichert strukturierte Daten
 - Ein Großteil wird implizit vom System mitgespeichert, kann also vom Dienst nicht gefälscht werden



systemd - Das Journal

Beispiel:

```
Di 2014-10-07 22:30:43.170745 CEST [s=95729ff918ce4...
  _TRANSPORT=syslog
  PRIORITY=6
  SYSLOG_FACILITY=4
  SYSLOG_IDENTIFIER=sshd
  MESSAGE=Received disconnect from
  2001:67c:670:100:1d::6: 11: disconnected by user
  _UID=999
  _GID=4001
  _COMM=sshd
  _CAP_EFFECTIVE=0
  _AUDIT_LOGINUID=999
  _SYSTEMD_OWNER_UID=999
  _SYSTEMD_SLICE=user-999.slice
  _BOOT_ID=f37591b9a4fd4418b18a27afaad0b0dd
  _MACHINE_ID=1f9112a51b3edc4d8ad934c800000007
  _HOSTNAME=dude
  SYSLOG_PID=24591
  _PID=24591
  _AUDIT_SESSION=38504
  _SYSTEMD_CGROUP=/user.slice/user-999.slice/session-38504.scope
  _SYSTEMD_SESSION=38504
  _SYSTEMD_UNIT=session-38504.scope
  _SOURCE_REALTIME_TIMESTAMP=1412713843170745
```



systemd - Das Journal

- Es wird immer das gesamte verfügbare Journal durchsucht
- --list-boots
- --since="2015-05-23 01:23:42" --until="2015-05-24"

- Bash: Autocompletion
 - hilfreich
 - kann sich zur Bremse entwickeln

- Zeige alle Werte von <FIELD>
 - journalctl -F_COMM
 - journalctl_PID=20400

- Container Support: -M | -m

```
bbu@pizza:~# journalctl -M foo -n 3
-- Logs begin at Wed 2015-03-18 13:12:27 CET, end at Thu 2015-03-19 15:42:26 CET.
Mar 18 15:42:06 foo systemd[1]: Started Cleanup of Temporary Directories.
Mar 19 15:42:26 foo systemd[1]: Starting Cleanup of Temporary Directories...
Mar 19 15:42:26 foo systemd[1]: Started Cleanup of Temporary Directories.
```



systemd - Das Journal - Neuigkeiten

- systemd-journal-gatewayd
 - syslog, verbose, binary, JSON
 - http GET (pull model)
 - http POST (push model)
 - Cursors → remember, continue
- Coredump Support
 - coredumpctl :-D
 - stack trace statt coredump im Log
 - Default Verzeichnis für Coredumps
- Journal Management
 - --vacuum-size=
 - --vacuum-time=
- seamless restart
 - journald übergibt seine offenen FD nun an PID 1
 - ... und bekommt sie bei restart wieder zurück



systemd - nspawn

- Eigentlich nur ein simpler Test-Container
- systemctl / machinectl / journalctl
- Guter Ersatz für chroot, ~500 Zeilen

```
bbu@pizza:~# systemctl status systemd-nspawn@fedora21
● systemd-nspawn@fedora21.service - Container fedora21
   Loaded: loaded (/usr/lib/systemd/system/systemd-nspawn@.service; enabled)
   Active: active (running) since Wed 2015-03-18 15:27:32 CET; 1 day 10h ago
     Docs: man:systemd-nspawn(1)
 Main PID: 702 (systemd-nspawn)
   Status: "Container running."
   CGroup: /system.slice/system-systemd\x2dnspawn.slice/systemd-nspawn@fedora21.service
           └─702 /usr/bin/systemd-nspawn --quiet --keep-unit --boot --link-journal=gu...
              └─776 /usr/lib/systemd/systemd
                 system.slice
                   dbus.service
                     └─1771 /bin/dbus-daemon --system --address=systemd: --nofork --nopic...
                   systemd-journald.service
                     └─888 /usr/lib/systemd/systemd-journald
                   systemd-logind.service
                     └─1766 /usr/lib/systemd/systemd-logind
                   console-getty.service
                     └─2121 /sbin/agetty --noclear --keep-baud console 115200 38400 9600 vt102

Mar 18 15:28:12 pizza systemd-nspawn[702]: Starting Update UTMP about System Runlevel ...
Mar 18 15:28:12 pizza systemd-nspawn[702]: [ OK ] Started Update UTMP about System ...
Mar 18 15:28:15 pizza systemd-nspawn[702]: Fedora release 21 (Twenty One)
Mar 18 15:28:15 pizza systemd-nspawn[702]: Kernel 3.18.9-200.fc21.x86_64 on an x86_64 ...
```



systemd - nspawn

- Interessante Neuigkeiten bei nspawn
 - Shares per overlays im Container
 - btrfs Integration: Snapshots
 - Templates
 - Flüchtige VM Images: --ephemeral
 - foo | systemd-nspawn | bar
- systemd-importd
 - gpg1 / gpg2 support
 - Import: raw, tar, qcow2
 - Export: tar, qcow2
 - Auch hier: btrfs Integration
- Achtung! Baustelle!



Dinge, die man sich anschauen sollte...





systemd – was ist los auf dem System?

```
root@machine:~ systemctl status
```

- machine

```
  State: running
  Jobs: 0 queued
  Failed: 0 units
  Since: Mon 2015-05-07 05:35:11 CET; 2min 27s ago
  CGroup: /
    └─1 /sbin/init
       └─system.slice
          └─dbus.service
             └─112 /usr/bin/dbus-daemon --system --address=systemd: [...]
          └─system-serial\x2dgetty.slice
             └─serial-getty@ttyLP0.service
                └─117 -sh
                   └─134 systemctl status
                      └─135 less
          └─systemd-journald.service
             └─59 /lib/systemd/systemd-journald
          └─systemd-logind.service
             └─114 /lib/systemd/systemd-logind
          └─systemd-udevd.service
             └─82 /lib/systemd/systemd-udevd
          └─NetworkManager.service
             └─125 /usr/sbin/NetworkManager --no-daemon [...]
```



systemd - Dokumentation

- Manpages & Infos
 - <http://0pointer.de/blog/projects/why.html>
 - <http://www.freedesktop.org/software/systemd/man/>
- `systemctl help some.service`
- `systemctl status some.service` + modernes Terminal

```
bbu@hobbes ~$ systemctl status systemd-logind
systemd-logind.service - Login Service
Loaded: loaded (/lib/systemd/system/systemd-logind.service; static)
Active: active (running) since Mo 2015-03-02 20:08:38 CET; 2 weeks 1 days ago
Docs: man:systemd-logind.service(8)
      man:logind.conf(5)
      http://www.freedesktop.org/wiki/Software/systemd/logind
      http://www.freedesktop.org/wiki/Software/systemd/multiseat
Main PID: 8564 (systemd-logind)
Status: "Processing requests..."
CGroup: /system.slice/systemd-logind.service
        └─8564 /lib/systemd/systemd-logind
```



systemd - Übergangslösungen

- In Debian funktionieren die alten init-scripte "noch" direkt
 - ... aber das kann schnell vorbei sein → siehe fedora
- Beispiel: ab systemd-220 nur noch statisches mapping der init level 2, 3, 4 -> multi-user.target / 5 -> graphical.target

```
root@hobbes:~# service avahi-daemon status
Redirecting to /bin/systemctl status avahi-daemon.service
● avahi-daemon.service - Avahi mDNS/DNS-SD Stack
   Loaded: loaded (/lib/systemd/system/avahi-daemon.service; enabled)
   Active: active (running) since Mo 2015-03-02 23:27:11 CET;          \
           2 weeks 1 days ago
   Main PID: 5385 (avahi-daemon)
   Status: "avahi-daemon 0.6.31 starting up."
   CGroup: /system.slice/avahi-daemon.service
           └─5385 avahi-daemon: running [hobbes.local]
             └─5387 avahi-daemon: chroot helper
```

- Am besten gleich an systemctl gewöhnen!
- Es gibt viele nützliche Kleinigkeiten zu entdecken
z.B. `systemctl {enable,disable,mask} --now <some.service>`



systemd - units

- Unit Debugging
 - systemd-analyze verify

```
~# cat motp.service
[Unit]
Description=Master of the Penguins Manager
After=syslog.target

[service]
Type=dbus
BusName=org.freedesktop.MotpManager1
ExecStart=/usr/sbin/MotpManager
StandardError=null
```

```
~# systemd-analyze verify ./motp.service
[/root/motp.service:5] \
Unknown section 'service'. Ignoring.
```



systemd - Debug

- Falls Start oder Shutdown „hängt“
 - 7 x CTRL-ALT-DEL in 2s
- Debug auf der Kernel Commandline aktivieren!
 - `systemd.log_level=debug`
 - `systemd.log_target=kmsg`
 - `log_buf_len=1M`
- Auch noch interessant:
 - `systemd.mask=`
 - `systemd.wants=`
 - `systemd.debug-shell`
- Debug auf Serieller Console
 - `systemd.log_level=debug`
 - `systemd.log_target=console`
 - `console=ttyS0,38400`



systemd - Kernel Commandline Optionen

- `systemd.unit=`, `rd.systemd.unit=`, `systemd.dump_core=`
- `systemd.crash_shell=`, `systemd.crash_chvt=`, `systemd.confirm_spawn=`
- `systemd.show_status=`, `systemd.log_target=`, `systemd.log_level=`
- `systemd.log_color=`, `systemd.log_location=`
- `systemd.default_standard_output=`, `systemd.default_standard_error=`
- `systemd.setenv=`

- `systemd.restore_state=`
- `systemd.journald.forward_to_syslog=`
- `systemd.journald.forward_to_kmsg=`
- `systemd.journald.forward_to_console=`
- `systemd.journald.forward_to_wall=`
- `vconsole.keymap=`, `vconsole.keymap.toggle=`, `vconsole.font=`
- `vconsole.font.map=`, `vconsole.font.unimap=`

- `systemd.gpt_auto=`, `rd.systemd.gpt_auto=`



systemd – Session-Bus

- Seit systemd-221:
 - stable sd-bus API
 - kdbus ist nicht mehr compile-time optional
 - kdbus kann weiterhin deaktiviert werden
- Debugging im Zusammenhang mit dbus
 - busctl



systemd - Stolperfallen

- Mehr Magie im System
 - Instanzierte Services
 - Beispiel: ttys
 - Impliziter Automounter für eine Menge Filesysteme
 - `sysctl.conf.d`
- shutdown
 - `-H` ist jetzt wirklich wieder "halt"
 - `-P` ist jetzt wirklich "poweroff"
- Es wird hinter den Prozessen aufgeräumt
 - `cgroups (!)`
 - Bug oder Feature:
Alle zu einer ssh session gehörenden Prozesse werden beim logout gekillt...
- Realtime Budget für bestimmt Prozesse?



systemd - autofs integration

Quelle (/etc/fstab):

```
nfssrv:/path/ /target nfs4 noauto,x-systemd.automount,defaults
```

```
root@dude:~# systemctl status ptx-work-himalia.automount
```

```
● ptx-work-himalia.automount
  Loaded: loaded (/etc/fstab)
  Active: active (running) since Mi 2015-01-21 10:53:26 CET;
          1 months 23 days ago
    Where: /ptx/work/himalia
     Docs: man:fstab(5)
           man:systemd-fstab-generator(8)
```

```
Mär 15 21:50:38 dude systemd[1]:
```

```
Got automount request for /ptx/work/himalia, triggered by 31789 (ls)
```

ab systemd-220:

- TimeoutIdleSec=60
- x-systemd.idle-timeout=60
- x-systemd.requires=



systemd - ditjes un' datjes

- `systemctl --failed`
- `systemctl kill -s HUP --kill-who={main,control,all} <service>`
- `systemd-cgls` \
- `systemd-analyze` / Analyse
- `systemctl cat` \
- `systemctl edit`) Service Unit Management
- `systemd-delta` /
- `systemd-escape` \
- `systemd-inhibit`) Scripting
- `systemd-notify` /
- `systemd-run` /



systemd - Zeitmanagement

- systemd-timesyncd Dump / Restore für Systeme ohne RTC
- timedatectl praktisches Helferlein

```
events@calvin:~$ timedatectl
Local time: Do 2015-04-02 12:56:51 CEST
  Universal time: Do 2015-04-02 10:56:51 UTC
    RTC time: Do 2015-04-02 10:56:51
      Time zone: Europe/Berlin (CEST, +0200)
        NTP enabled: no
NTP synchronized: yes
  RTC in local TZ: no
    DST active: yes
Last DST change: DST began at
                  So 2015-03-29 01:59:59 CET
                  So 2015-03-29 03:00:00 CEST
Next DST change: DST ends (the clock jumps one hour backwards) at
                  So 2015-10-25 02:59:59 CEST
                  So 2015-10-25 02:00:00 CET
```



systemd – DNS / Netzwerk

- systemd-hostnamed Einheitliche Repräsentation der Host-Details
- systemd-networkd Container Support, dhcp / dhcpv6 Server
- systemd-resolved Caches per interface, LLMNR, dnssec
- nssmyhostname /etc/hosts wird ggf. nicht mehr gebraucht
- systemd-netdev Virtual Device Management

```
bbu@pizza:~$ cat /usr/lib/systemd/network/80-container-ve.network
[Match]
Name=ve-*
Driver=veth

[Network]
Address=0.0.0.0/28
IPv4LL=yes
DHCPServer=yes
```

- IP Forwarding / Masquerading
- Application Firewall Support (via cgroup netclass)
- DHCPd
- PPPoE



systemd - auch noch einen Blick wert

- GPT Autodiscovery
 - Automatische Erkennung der Partitionen
 - Ziel: Boot ohne /etc/fstab
- <http://www.freedesktop.org/wiki/Specifications/BootLoaderSpec>
- Gummiboot Integration: „sd-boot“
 - "Trusted Boot" ohne M\$-Keys in der Hardware
 - Reboot to EFI Firmware
 - EFI Firmware Stub -> Signierte Komplett-Images
 - bootctl
- console support
 - Text Console in Userspace / DRM
 - skalierbar
 - Unicode Support
 - Multiseat Support



**Guter Einstieg ins Thema:
The systemd for Administrators Blog Series
<http://www.freedesktop.org/wiki/Software/systemd/>**



Vielen Dank für Ihre Aufmerksamkeit!

`systemctl [hibernate | suspend | hybrid-sleep]`



<http://public.pengutronix.de/slac-2015/>



Kontakt

Pengutronix
Peiner Strasse 6-8
31137 Hildesheim
Germany

Phone: +49 - 51 21 / 20 69 17 - 0

Fax: +49 - 51 21 / 20 69 17 - 9

Mail: sales@pengutronix.de

Web: <http://www.pengutronix.de>

