



Umstellung auf Systemd

The name "Gentoo" and the "g" logo are currently trademarks of Gentoo Foundation, Inc.



Übersicht

- aixTeMa GmbH
- *systemd*
- *dnotify* Umstellung auf *systemd*
- *systemd-nspawn* ist *chroot* unter *steroids*
- Verhalten nach der Umstellung auf *systemd*
- Bekannte Probleme
- Fazit

Was wir anbieten:

- Logistik, Technische Inbound Hotline, Reparatur-Service, IT-Lösungen
- *Asterisk* Telefonanlage
- 20 Server + 15 virtualisierte Server
- 25 Linux Arbeitsplätze (Hotline)
- 90 Rechner und 13 Notebooks mit Linux und Windows

Was vor dem emerge zu ändern ist:

- In der `/etc/portage/make.conf` zu den USE Flags `systemd udev gudev` hinzufügen.
- In der `/etc/portage/package.use` alle Einträge von `sys-fs/udev virtual/libudev dev-libs/libgudev` und `virtual/libgudev` entfernen.
- `static-libs` Einträge in der `make.conf` und `package.use` in `-static-libs` ändern.

```
emerge -C sys-fs/udev virtual/libudev dev-libs/libgudev virtual/libgudev
```

```
emerge -1av systemd
```

```
emerge -uvDN systemd
```

```
emerge @preserved-rebuild
```

- **mtab ändern:**

```
rm /etc/mtab && ln -sf /proc/self/mounts /etc/mtab
```

- **Booten mit systemd:**

- **Raspberry PI/Odroid:**

init=/usr/lib/systemd/systemd in der /boot/cmdline.txt einfügen

- **Grub:**

```
/boot/grub/grub.conf
```

```
kernel /vmlinuz root=/dev/sda2 init=/usr/lib/systemd/systemd
```

- **Grub2:**

```
/etc/default/grub
```

```
GRUB_CMDLINE_LINUX="init=/usr/lib/systemd/systemd"
```

systemd Vor- und Nachteile

Vorteile:

- Start-, Abschaltzeit Reduzierung

Beispiel Raspberry PI 2

udev 57 Sekunden 17 Sekunden

systemd 17 Sekunden 7 Sekunden

Nachteile:

- Nicht jedes Programm hat schon eine *.service* Datei
- MD *raids* starten langsamer

Lösungen:

- Es gibt viele Beispiele für eigene *.service* Dateien
- *networkd* abschalten und *netctl* verwenden

Vergleich von sshd *init.d/systemd*

```
vi /etc/init.d/sshd
```

```
extra_commands="checkconfig gen_keys"
extra_started_commands="reload"
depend() {
    use logger dns
    need net
}
SSHD_CONFDIR=${SSHD_CONFDIR:-/etc/ssh}
SSHD_PIDFILE=${SSHD_PIDFILE:-/var/run/${SVCNAME}.pid}
SSHD_BINARY=${SSHD_BINARY:-/usr/sbin/sshd}
checkconfig() {
    if [ ! -d /var/empty ] ; then
        mkdir -p /var/empty || return 1
    fi
    if [ ! -e "${SSHD_CONFDIR}/sshd_config ] ; then
        error "You need an ${SSHD_CONFDIR}/sshd_config
file to run sshd"
        error "There is a sample file in
/usr/share/doc/openssh"
        return 1
    fi
}
....
```

Und so weiter.

```
vi /usr/lib/systemd/system/sshd.service
```

```
[Unit]
Description=OpenSSH server daemon
After=syslog.target network.target auditd.service

[Service]
ExecStartPre=/usr/bin/ssh-keygen -A
ExecStart=/usr/sbin/sshd -D -e
ExecReload=/bin/kill -HUP $MAINPID

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

Das war schon alles!

dnotify Umstellung auf systemd

/etc/init.d/dnotify-example

```
start() {
    ebegin "start dnotify-example"
    start-stop-daemon --start --background --pidfile --pidfile
/var/run/dnotify-example.pid --exec /root/bin/dnotify-example.sh
    eend $? " failed to start"
}
stop() {
    if [ -f /var/run/dnotify-example.pid ]; then
    let PID=$(cat /var/run/dnotify-example.pid)+1
    echo $PID > /var/run/dnotify-example.pid
    fi
    ebegin " stop dnotify-example "
    start-stop-daemon --stop --quiet --pidfile dnotify-
example.pid
    eend $? " failed to kill"
```

dnotify-example.service

```
[Unit]
Description=dnotify-logistik.service

[Service]
Type=forking
User=root # (root tut gut)
ExecStart=/root/bin/dnotify-example.sh

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

rc-status Auszug:

dnotify-example [**crashed**]

Neustart nur via **kill** auf die *pid* möglich!

systemctl Auszug:

dnotify-example loaded active
running dnotify-example.service

start ,stop ,restart möglich!

systemd-nspawn ist chroot unter steroids

- [Systemd-nspawn ausführlich](#) (youtube Video)
- *systemd-nspawn* nur für die Entwicklung / Paketbau kein Live Betrieb
- Eigener *init* zum starten und stoppen von Prozessen
- Bauen und konfigurieren von kompletten Systemen (z.B. Hotline System)
- Kein *mount* von *proc sys dev* mehr nötig, es reicht ein: ***systemd-nspawn -bD /path/to/dir***
- Bau vom **Kernel nicht möglich**, da *mknod* verwendet wird, könnte aber mit etwas wie ***fakebuild*** bei Debian gelöst werden

- Wir hatten ein paar Probleme mit *networkd* und Xen
- Lösung laut ARCH: einfach auf *netctl* umstellen
- Ein Manko gibt es:
Ist bei einem Systemstart kein Netzwirkabel eingesteckt, muss man den Dienst entweder von Hand starten, wenn ein Kabel eingesteckt wurde, oder es per *anacron* und/oder *crontab* forcieren.

- Erster Systemstart hängt.
Dies war bei uns das erste mal am 14.08.2013.
- Hostname wird nicht mehr aus `conf.d/hostname` übernommen

```
hostnamectl set-hostname $hostname
```

- Tastatur Layout per Default auf Englisch

```
localectl set-locale LANG="de_DE.utf8"  
localectl set-keymap=de_DE.latin1-nodeadkeys
```

- Zeitzone setzen

```
timedatectl set-timezone Europe/Berlin  
rm /etc/timezone /etc/localtime  
ln -s /usr/share/zoneinfo/Europe/Berlin /etc/localtime  
echo "Europe/Berlin" > /etc/timezone
```

- **Der PC ist irgendwie aus, da autosuspend an ist!**

```
vi /etc/systemd/logind.conf  
HandleLidSwitch=ignore
```

Danach Neustarten !!!

- **Das var Verzeichnis läuft voll,
und es sind diesmal keine MySQL Logs!**

```
vi /etc/systemd/journald.conf  
Compress=yes  
RuntimeMaxUse=5M  
RuntimeKeepFree=100M
```

- ***ntpd* und *ntp-client* laufen nicht beim Systemstart mit systemd**

Den ntpd benutzen und schon läuft alles wieder.

- **Selbst geschriebene anacron Jobs laufen seit *systemd* Version 221 nicht mehr richtig**

Wir verweisen hier auf:

<http://jason.the-graham.com/2013/03/06/how-to-use-systemd-timers/>

- **Script um einige Einstellungen schon nach dem Neustart auszuführen:**

<http://it-shamans.eu/Downloads/systemd/to-do-systemd.sh>

- **Wer schnell sein will muss Skripte anpassen.**
- **Wann lohnt sich die Umstellung?**
Um so langsamer die Systeme (ARM, Atom) oder Speichermedien (SDcard, HDD per USB) sind, desto eher lohnt sich das!
- **Bei geplantem System Upgrade**
Bei einem Webserver mit Apache2, mysql, ssh und ftp
ca. 30 Minuten mehr Aufwand.
Vorausgesetzt man hat die dazugehörigen *.service* Dateien.

Vielen Dank für Ihre Aufmerksamkeit!

Weitere Infos zu *systemd* auf <https://it-shamans.eu>
unter **Suche** *systemd* eingeben!