

Systemd-Übung

Quelle: Text aus der c't 2011, Heft 13, Seite: 177!



1. Sind nach dem Umstieg auf systemd die Funktions- und Dienstaufrufe von Sys-V-init oder runit noch funktionsfähig?

Nein! – Mit „*/etc/init.d/Dienstname start*“ lassen sich nicht mehr alle Daemons starten! Bei systemd startet man einen Dienst durch: „*Dienstname start*“, oder: „*systemctl start Dienstname*“.

2. Wie lautet der wichtigste Befehl, den das systemd-System zur Verwaltung anbietet?

3. Was ist bei systemd eine „UNIT“ – wie z.B. home.mount, boot.mount, bluetooth.service, getty@tty1.service, kdm.service, mysqld.service oder systemd-journald.service?

Es handelt sich um einmalige Systemaufrufe bzw. Systemdienste, die im Hintergrund laufen und dabei systemrelevante Aufgaben übernehmen oder zum Startzeitpunkt übernommen haben und bereits regulär wieder beendet wurden.

4. Wie lassen sich ausschließlich die laufenden System-Dienste (service-units) bei systemd abfragen?

systemctl --type=service

5. Wie kann man nachsehen welche „units“ installiert wurden?

systemctl list-unit-files

6. Mit welchem Systemaufruf in der bash lässt sich der Systemzustand in einer temporären Datei (z. B. /tmp/zustand.txt) speichern?

systemctl status > /tmp/zustand.txt

oder: systemctl > /tmp/zustand.txt

7. Mit welchem Systemaufruf kann man die beendeten service-units zur Anzeige bringen? *systemctl --type=service | grep exited*

8. Was zeigt nachfolgender Aufruf: *systemctl --type=service | grep running*?

Die gerade laufenden System-Units (System-Dienste=Service)!

9. Was bedeutet ein „failed“ in der fünften Spalte der Anzeige, wenn man nachfolgenden Befehl eingibt: *systemctl --type=service*?

Der Service wurde abgebrochen bzw. er konnte nicht richtig gestartet („hochgefahren“) werden.

10. Was bewirkt der Befehl: *systemctl status ntpd.service*?

s wird der Zustand des networktime-service abgefragt. Dabei erfolgt jedoch zusätzlich die Ausgabe eines Protokolls, das in roter Farbe ausgibt, welche Dienste wann mit welchem Fehlercode einen Abbruch erfahren hat.



11. Wo liegen die Konfigurationsdateien zum Erzeugen einer solchen „unit“ ?

n „/lib/systemd/system“ oder in „/etc/systemd/system“, wobei die zweite einen Vorrangstellung einnimmt.

12. Wie sieht das Grundgerüst einer „unit“ aus?

[Unit]

Description=.....

Service]

ExecStart=Befehl mit Parametern

ExecStpop=Befehl zum stoppen des Dienstes

[Install]

WantedBy=.....

e nach Bedarf, kann das Gerüst noch erweitert werden! Es gibt unterschiedliche Betriebszustände z.B. single-user.target, multi-user.target usw. Diese sind mit den runleveln des Sys-V-init vergleichbar!

13. Was bewirkt der Befehl: „ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target“ ?

Es wird ein symbolischer Link erzwungen (f=force) bzw. überschrieben, falls dieser bereits vorhanden ist, der mit dem Namen: default... versehen wird. Dieser Link stellt dann den voreingestellten Betriebsmodus her, indem er auf multi-user.target verweist, also den Betriebsmodus Multiuser = mehrere Konsolen (tty) ohne graphische Oberfläche startet

14. Wie wechselt man den Betriebsmodus aus dem laufenden Betrieb in den reinen Multiuser-Konsolenmodus ohne

Grafik? **systemctl isolate rescue.target**

15. Sie möchten überprüfen welche Dienste oder andere targets für einen bestimmten Zielbetrieb (target) initialisiert werden sollen und ihren Betrieb

aufnehmen! **Systemctl show -p Wants multi-user.target.**

16. targets können nacheinander oder zeitlich parallel verarbeitet werden. Was bedeuten in diesem Zusammenhang die Angaben:

requires = setzt den Start eines Dienstes oder anderen targets voraus!

After = setzt voraus, dass der Dienst bereits vollständig gestartet wurde und den

Wants Betrieb aufgenommen hat. Erst danach geht es weiter.

17. Wie kann ein Dienst dauerhaft auf „ein“ bzw. auf „aus“ gesetzt werden, so dass diese Einstellungen nach dem nächsten Neustart (nicht) dauerhaft zur Anwendung kommen?

systemctl enable ntpd.service

oder: **systemctl disable ntpd.service**

sysd

systemd-ui: systemadm

Grafik vom Systemd-Startvorgang erzeugen: **systemd-analyze plot > plot.svg**

Service Related Commands

Comments	SysVinit	Systemd
Start a service	service dummy start	systemctl start dummy.service
Stop a service	service dummy stop	systemctl stop dummy.service
Restart a service	service dummy restart	systemctl restart dummy.service
Reload a service	service dummy reload	systemctl reload dummy.service
Service status	service dummy status	systemctl status dummy.service
Restart a service if already running	service dummy condrestart	systemctl condrestart dummy.service
Enable service at startup	chkconfig dummy on	systemctl enable dummy.service
Disable service at startup	chkconfig dummy off	systemctl disable dummy.service
Check if a service is enabled at startup	chkconfig dummy	systemctl is-enabled dummy.service
Create a new service file or modify configuration	chkconfig dummy --add	systemctl daemon-reload

Note : New version of systemd support "systemctl start dummy" format.

Runlevels

Comments	SysVinit	Systemd
System halt	0	runlevel0.target, poweroff.target
Single user mode	1, s, single	runlevel1.target, rescue.target
Multi user	2	runlevel2.target, multi-user.target
Multi user with Network	3	runlevel3.target, multi-user.target
Experimental	4	runlevel4.target, multi-user.target
Multi user, with network, graphical mode	5	runlevel5.target, graphical.target
Reboot	6	runlevel6.target, reboot.target
Emergency Shell	emergency	emergency.target
Change to multi user runlevel/target	telinit 3	systemctl isolate multi-user.target (OR systemctl isolate runlevel3.target)
Set multi-user target on next boot	sed s/^id:.*initdefault:/id:3:initdefault:/	ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target
Check current runlevel	runlevel	systemctl get-default
Change default runlevel	sed s/^id:.*initdefault:/id:3:initdefault:/	systemctl set-default multi-user.target

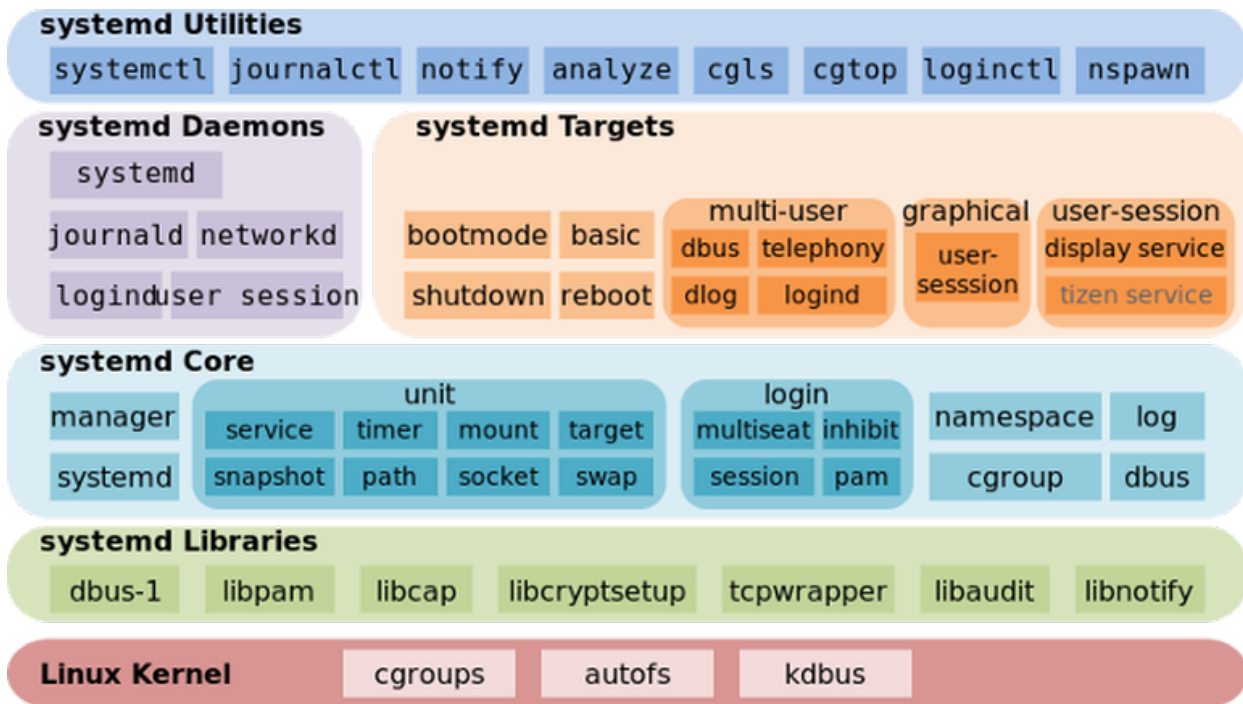
Miscellaneous Commands

Comments	SysVinit	Systemd
System halt	halt	systemctl halt
Power off the system	poweroff	systemctl poweroff
Restart the system	reboot	systemctl reboot
Suspend the system	pm-suspend	systemctl suspend
Hibernate	pm-hibernate	systemctl hibernate
Follow the system log file	tail -f /var/log/messages or tail -f /var/log/syslog	journalctl -f

Systemd New Commands

Comments	Systemd
Execute a systemd command on remote host	systemctl dummy.service start -H user@host
Check boot time	systemd-analyze or systemd-analyze time
Kill all processes related to a service	systemctl kill dummy
Get logs for events for today	journalctl --since=today
Hostname and other host related information	hostnamectl
Date and time of system with timezone and other information	timedatectl

If you're not a Linux or Unix developer, you've never heard of [systemd](#), the new Linux-specific system and service manager. In Linux developer circles, however, nothing else ticks off many programmers more than this replacement for the Unix and Linux's traditional [sysvinit daemon](#).



Many

Linux developers think systemd tries to do way, way too much for an init program.

Systemd provides a standard process for controlling what programs run when a Linux system boots up. While systemd is compatible with [SysV](#) and [Linux Standard Base \(LSB\) init scripts](#), systemd is meant to be a drop-in replacement for these older ways of getting a Linux system running.

Systemd, which was created by [Red Hat](#)'s Lennart Poettering and Kay Sievers, does more than start the core programs running. It also starts a journal of system activity, the network stack, a [cron](#)-style job scheduler, user logins, and many other jobs. That may sound good to you, but some developers *hate* it.

On the site [Boycott Systemd](#), the authors lash out at systemd stating:

"Systemd flies in the face of the Unix philosophy: 'do one thing and do it well,' representing a complex collection of dozens of tightly coupled binaries¹. Its responsibilities grossly exceed that of an init system, as it goes on to handle power management, device management, mount points, cron, disk encryption, socket API/inetd, syslog, network configuration, login/session management, readahead, GPT partition discovery, container registration, hostname/locale/time management, and other things. Keep it simple, stupid."

Because systemd puts so many of a program's eggs in one system basket, systemd's critics argue that "there are tons of scenarios in which it can crash and bring down the whole system. But in addition, this means that plenty of non-kernel system upgrades will now require a reboot. Enjoy your new Windows 9 Linux system!"

They go on to argue that systemd's journal files, which are stored in a binary format, are potentially corruptible. In addition, they find that systemd is incompatible with other members of the Unix operating system family. They also flaw it for its "monolithic, heavily desktop-oriented," design, which makes it a poor choice for many Linux use cases

[Poettering has addressed these concerns](#) many times since systemd appeared but the criticisms keep coming. What makes all this arguing over systemd especially odd is that, despite all this hate, it's been widely adopted. The [GNOME 3.8 desktop and newer now requires systemd](#). [Fedora](#), Red Hat's community Linux, was the first major distribution to start using it as a default. Since then, [Debian Linux](#), [openSUSE](#), and [Ubuntu](#) have all adopted systemd.

So what do Linux's leaders think of all this? I asked them and this is what they told me.

Linus Torvalds said:

"I don't actually have any particularly strong opinions on systemd itself. I've had issues with some of the core developers that I think are much too cavalier about bugs and compatibility, and I think some of the design details are insane (I dislike the binary logs, for example), but those are details, not big issues."

Theodore "Ted" Ts'o, a leading Linux kernel developer and a Google engineer, sees systemd as potentially being more of a problem. "The bottom line is that they are trying to solve some real problems that matter in some use cases. And, [that] sometimes that will break assumptions made in other parts of the system."

Another concern that Ts'o made — which I've heard from many other developers — is that the systemd move was made too quickly: "The problem is sometimes what they break are in other parts of the software stack, and so long as it works for GNOME, they don't necessarily consider it their responsibility to fix the rest of the Linux ecosystem."

This, as Ts'o sees it, feeds into another problem:

"Systemd problems might not have mattered that much, except that [GNOME](#) has a similar attitude; they only care for a small subset of the Linux desktop users, and they have historically abandoned some ways of interacting the Desktop in the interest of supporting touchscreen devices and to try to attract less technically sophisticated users. If you don't fall in the demographic of what GNOME supports, you're sadly out of luck. (Or you become a second class citizen, being told that you have to rely on GNOME extensions that may break on every single new version of GNOME.)"

Ts'o has an excellent point. [GNOME 3.x has alienated both users and developers](#). He continued, "As a result, many traditional GNOME users have moved over to Cinnamon, XFCE, KDE, etc. But as systemd starts subsuming new functions, components like network-manager will only work on systemd or other components that are forced to be used due to a network of interlocking dependencies; and it may simply not be possible for these alternate desktops to continue to function, because there is [no] viable alternative to systemd supported by more and more distributions."

What's Hot on ZDNet

- [2014: I'm still sick of Android](#)
- [2014: The year Microsoft lost my loyalty](#)
- [The fall of GPL and the rise of permissive open-source licenses](#)
- [The very best tech you can buy for \\$200 or less](#)

Of course, Ts'o continued, "None of these nightmare scenarios have happened yet. The people who are most stridently objecting to systemd are people who are convinced that the nightmare scenario is inevitable so long as we continue on the same course and altitude."

Ts'o is "not entirely certain it's going to happen, but he's afraid it will.

What I find puzzling about all this is that even though everyone admits that sysvinit needed replacing and many people dislike systemd, the distributions keep adopting it. Only a few distributions, including [Slackware](#), [Gentoo](#), [PCLinuxOS](#), and [Chrome OS](#), haven't adopted it.

It's not like there aren't alternatives. These include [Upstart](#), [runit](#), and [OpenRC](#).

If systemd really does turn out to be as bad as some developers fear, there are plenty of replacements waiting in the wings. Indeed, rather than hear so much about how awful systemd is, I'd rather see developers spending their time working on an alternative.

Related Stories:

- [After Linux civil war, Ubuntu to adopt systemd](#)
- [Debian inches towards new init system decision amid fallout](#)
- [Debian inches towards new init system decision amid fallout](#)

Musterlösung zur Systemd-Übung:

1. Sind nach dem Umstieg auf systemd die Funktions- und Dienstaufrufe von Sys-V-init oder runit noch funktionsfähig?

Nein! – Mit „/etc/init.d/Dienstname start“ lassen sich nicht mehr alle Daemons starten! Bei systemd startet man einen Dienst durch: „Dienstname start“, oder: „systemctl start Dienstname“.

2. Wie lautet der wichtigste Befehl, den das systemd-System zur Verwaltung anbietet?

Systemctl „viele Parameter“ nutze den Parameter: --help!

3. Was ist bei systemd eine „UNIT“ – wie z.B. home.mount, boot.mount, bluetooth.service, getty@tty1.service, kdm.service, mysqld.service oder systemd-journald.service?

Es handelt sich um einmalige Systemaufrufe bzw. Systemdienste, die im Hintergrund laufen und dabei systemrelevante Aufgaben übernehmen oder zum Startzeitpunkt übernommen haben und bereits regulär wieder beendet wurden.

4. Wie lassen sich ausschließlich die laufenden System-Dienste (service-units) bei systemd abfragen?

systemctl --type=service

5. Wie kann man nachsehen welche „units“ installiert wurden?

systemctl list-unit-files

6. Mit welchem Systemaufruf in der bash lässt sich der Systemzustand in einer temporären Datei (z. B. /tmp/zustand.txt) speichern?

systemctl status > /tmp/zustand.txt

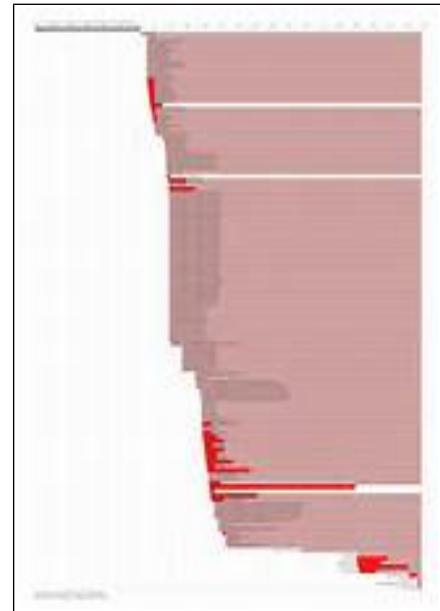
oder: systemctl > /tmp/zustand.txt

7. Mit welchem Systemaufruf kann man die beendeten service-units zur Anzeige bringen?

systemctl --type=service | grep exited

8. Was zeigt nachfolgender Aufruf: systemctl --type=service | grep running?

Die gerade laufenden System-Units (System-Dienste=Service)!



9. Was bedeutet ein „failed“ in der fünften Spalte der Anzeige, wenn man nachfolgenden Befehl eingibt: `systemctl --type=service`?

Der Service wurde abgebrochen bzw. er konnte nicht richtig gestartet („hochgefahren“) werden.

10. Was bewirkt der Befehl: `systemctl status ntpd.service`?

Es wird der Zustand des networktime-service abgefragt. Dabei erfolgt jedoch zusätzlich die Ausgabe eines Protokolls, das in roter Farbe ausgibt, welche Dienste wann mit welchem Fehlercode einen Abbruch erfahren hat.

11. Wo liegen die Konfigurationsdateien zum Erzeugen einer solchen „unit“ ?

Diese liegen unter: „/lib/systemd/system“ oder in „/etc/systemd/system“, wobei die zweite einen Vorrangstellung einnimmt.

12. Wie sieht das Grundgerüst einer „unit“ aus?

[Unit]

Description=.....

[Service]

ExecStart=Befehl mit Parametern

ExecStop=Befehl zum stoppen des Dienstes

[Install]

WantedBy=.....

Je nach Bedarf, kann das Gerüst noch erweitert werden! Es gibt unterschiedliche Betriebszustände z.B. single-user.target, multi-user.target usw. Diese sind mit den runleveln des Sys-V-init ergleichbar!

13. Was bewirkt der Befehl: „`ln -sf /lib/systemd/system/multi-user.target /etc/systemd/system/default.target`“ ?

Es wird ein symbolischer Link erzwungen (f=force) bzw. überschrieben, falls dieser bereits vorhanden ist, der mit dem Namen: default... versehen wird. Dieser Link stellt dann den voreingestellten Betriebsmodus her, indem er auf multi-user.target verweist, also den Betriebsmodus Multiuser = mehrere Konsolen (tty) ohne graphische Oberfläche startet.

14. Wie wechselt man den Betriebsmodus aus dem laufenden Betrieb in den reinen Multiuser-Konsolenmodus ohne Grafik?

systemctl isolate rescue.target

15. Sie möchten überprüfen welche Dienste oder andere targets für einen bestimmten Zielbetrieb (target) initialisiert werden sollen und ihren Betrieb aufnehmen!

Systemctl show -p Wants multi-user.target.

16. targets können nacheinander oder zeitlich parallel verarbeitet werden. Was bedeuten in diesem Zusammenhang die Angaben:

Requires = setzt den Start eines Dienstes oder anderen targets voraus!

After = setzt voraus, dass der Dienst bereits vollständig gestartet wurde und den Betrieb aufgenommen hat. Erst danach geht es weiter.

Wants = ist ein abgeschwächtes Require

17. Wie kann ein Dienst dauerhaft auf „ein“ bzw. auf „aus“ gesetzt werden, so dass diese Einstellungen nach dem nächsten Neustart (nicht) dauerhaft zur Anwendung kommen?

systemctl enable ntpd.service

oder: ***systemctl disable ntpd.service***

syst

systemd-ui: systemadm

Grafik vom Systemd-Startvorgang erzeugen:

systemd-analyze plot > plot.svg