

# Systemd und System-Initialisierung



# Inhaltsverzeichnis

-

# Systemd - Historie

Systemd entstand als Reaktion auf die Komplexität und die langen Startzeiten des traditionellen `SysVinit`-Systems. Ein weiterer Grund war die mangelnde Parallelisierung des Startvorgangs.

Konzepte und Ideen von Systemd stammen aus *Launchd* von *macOS* und *SMF* (*Service Management Facility*) von *Solaris*. Lange Zeit war *Upstart* von *Ubuntu* ein Konkurrent von Systemd.

- Systemd wurde von Lennart Poettering und Kay Sievers entwickelt und erstmals 2010 in Fedora 15 eingeführt.
- In den folgenden Jahren wurde Systemd in vielen Linux-Distributionen als Standard-Init-System eingeführt (*Fedora, openSUSE, Arch Linux, Mandriva, Mageia, RHEL (Red Hat Enterprise Linux), SLES (SUSE Linux Enterprise Server)*).
- *Ubuntu* wechselte 2015 vom selbst entwickelten Upstart-Init-System zu Systemd. Auch *Debian (Jessie)* führt 2015 Systemd ein.
- Damit wurde Systemd zum Standard-Init-System in den meisten Linux-Distributionen.

# Systemd - Aufgaben

Systemd ist ein System- und Service-Manager für Linux, der das System Initialisiert und danach im laufenden Betrieb verwaltet. Systemd hat folgende Aufgaben:

- Steuerung und Protokollierung des Systemstarts (nach dem Laden des Kernels) und des Shutdowns
- Bereitstellen der Hardware
- Ein- und Aushängen der Dateisysteme (auch Automounts)
- Starten und Beenden der Systemdienste

- Bereitstellen von Sockets zur Kommunikation zwischen Systemprozessen
- Steuerung von User-Sessions (vom Login bis zum Logout)
- Logging und Protokollierung
- Ressourcenkontrolle
- Zeitgesteuerte Aufgaben

Mehr dazu in *L07a-Systemd-Vorläufer-Konkurrenten-Ideengeber*

# Systemd-Unit-Types (Aufgabengruppen)

- `service` - Systemdienste
- `socket` - Sockets
- `device` - Hardwaregeräte, repräsentiert durch Dateien im `/dev` - Dateisystem
- `mount` - Eingehängte Dateisysteme
- `automount` - Automatisch eingehängte Dateisysteme
- `target` - Gruppierung anderer Units

- `snapshot` - gespeicherte Systemzustände
- `timer` - Zeitgesteuerte Units
- `path` - Dateisystem-Änderungen
- `slice` - Ressourcenkontrolle
- `scope` - Prozessgruppen
- `swap` - Swap-Dateien



```
hermann@debian:~$ # show available 'systemd' unit types
hermann@debian:~$ systemctl list-units --type=help | nl -ba
  1 Available unit types:
  2 service
  3 mount
  4 swap
  5 socket
  6 target
  7 device
  8 automount
  9 timer
 10 path
 11 slice
 12 scope
```

Jeder Unit-Typ kann mit dem Parameter `--type=unit-type` angegeben werden, um die Units dieses Typs zu listen.

## Units vom Typ `service`

```
hermann@debian:~$ systemctl list-units --type=service \
> ssh.service cron.service smbd.service nmbd.service
UNIT          LOAD    ACTIVE SUB    DESCRIPTION
cron.service  loaded active running Regular background program processing daemon
nmbd.service  loaded active running Samba NMB Daemon
smbd.service  loaded active running Samba SMB Daemon
ssh.service   loaded active running OpenBSD Secure Shell server
```

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type. ...

Hier wurden vier Units vom Typ `service` angegeben. Werden keine Services angegeben, werden alle gelistet.

## Units vom Typ **mount**

```
hermann@debian:~$ systemctl list-units --type=mount -- -.mount \
> boot-efi.mount dev-hugepages.mount dev-mqueue.mount
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
-.mount                            loaded active mounted Root Mount
boot-efi.mount                     loaded active mounted /boot/efi
dev-hugepages.mount                loaded active mounted Huge Pages File System
dev-mqueue.mount                   loaded active mounted POSIX Message Queue File System
```

LOAD = Reflects whether the unit definition was properly loaded.  
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.  
SUB = The low-level unit activation state, values depend on unit type. ...

Hier wurden vier Units vom Typ **mount** angegeben. Werden keine Mounts angegeben, werden alle gelistet.

# Erläuterung der Unit-Types

Die Erläuterung der Unit-Types gibt einen Überblick über die verschiedenen Aufgaben, für die Systemd zuständig ist.

## Unit-Type **service** - Systemdienste

- Systemdienste sind Programme, die im Hintergrund laufen und bestimmte Aufgaben erfüllen.
- Systemdienste können von Systemd gestartet, gestoppt, neugestartet und neu geladen werden.
- Systemdienste können automatisch beim Systemstart gestartet werden.

## Unit-Type `socket` - Sockets

- Sockets (Unix Domain Sockets) sind Kommunikationsendpunkte, die von Prozessen auf dem lokalen System genutzt werden, um miteinander zu kommunizieren.
- Sockets können von Systemd erstellt und verwaltet werden.
- Sie sind Kommunikationsendpunkte für Systemdienste, die auch Daten puffern können. Z.B. `systemd-journald.socket`. Über diesen Socket können andere Dienste ihre Log-Daten an `systemd-journald` senden.

## Unit-Type `device` - Hardwaregeräte

- Hardwaregeräte sind Geräte, die an das System angeschlossen oder eingebaut sind und von Systemd erkannt, initialisiert und verwaltet werden.
- Hardwaregeräte werden durch Dateien im `/dev`-Dateisystem repräsentiert.
- Die Dateien im `/dev`-Dateisystem werden teils dynamisch erstellt. Z.B. wird ein USB-Stick beim Einstecken dynamisch erkannt und ein Device-File im `/dev`-Verzeichnis erstellt. Oder ein Pseudo-TTY wird beim Anmelden eines Benutzers mit `ssh` dynamisch erstellt und beim Abmelden wieder gelöscht.

## Unit-Type `mount` - Eingehängte Dateisysteme

- Dateisysteme (physikalische oder virtuelle) werden von Systemd eingehängt und verwaltet.
- Beim Systemstart werden die Dateisysteme eingehängt und beim Shutdown wieder ausgehängt.
- In `/etc/fstab` werden die Dateisysteme und ihre Mount-Optionen definiert. Systemd liest diese Datei und hängt die Dateisysteme ein.
- Wird `/etc/fstab` (vom Administrator) geändert, muss Systemd neu geladen werden, damit die Änderungen wirksam werden (`systemctl daemon-reload`).



# Unit-Type **automount** - Automatisch eingehängte Dateisysteme

- Automounts sind Dateisysteme, die automatisch eingehängt werden, wenn auf sie zugegriffen wird.
- Automounts werden von Systemd verwaltet.
- Automounts sind nützlich, wenn auf ein Dateisystem nur selten zugegriffen wird. Das Dateisystem wird erst eingehängt, wenn darauf zugegriffen wird, und wird nach einer gewissen Zeit der Inaktivität automatisch wieder ausgehängt.

## Unit-Type `target` - Gruppierung anderer Units

- Targets sind Gruppen von Units, die zusammen gestartet oder gestoppt werden.
- Targets entsprechen (ganz grob) den Runlevels des traditionellen `SysVinit`-Systems.

- Targets definieren bestimmte Systemzustände, z.B.
  - `multi-user.target` - System ist im Multi-User-Modus
  - `graphical.target` - System ist im Grafik-Modus
  - `bluetooth.target` - System ist im Bluetooth-Modus
  - `network.target` - System ist im Netzwerk-Modus
  - `single.target` - System ist im Single-User-Modus
  - `default.target` - Standard-Target, Ziel-Target beim Start des Systems
  - etc.

## Unit-Type **snapshot** - Gespeicherte Systemzustände

- Snapshots sind gespeicherte Systemzustände, die von Systemd erstellt und verwaltet werden.
- Snapshots können erstellt werden, um den Systemzustand zu sichern, bevor Änderungen vorgenommen werden.
- Snapshots können verwendet werden, um einen früheren Systemzustand wiederherzustellen.

## Unit-Type **timer** - Zeitgesteuerte Units

- Timer sind Units, die zu bestimmten Zeiten oder in bestimmten Intervallen ausgeführt werden.
- Timer können verwendet werden, um bestimmte Aufgaben zeitgesteuert auszuführen.

## Unit-Type **path** - Dateisystem-Änderungen

- Path-Units überwachen Dateisystem-Änderungen und führen eine Aktion aus, wenn eine bestimmte Datei oder ein bestimmtes Verzeichnis geändert/gelöscht/hinzugefügt wird.

## Unit-Type **slice** - Ressourcenkontrolle

- Slices sind Gruppen von Prozessen, die gemeinsam Ressourcen nutzen.
- Slices können verwendet werden, um die Ressourcennutzung von Prozessen zu kontrollieren oder zu begrenzen.

## Unit-Type **scope** - Prozessgruppen

- Scopes sind Gruppen von Prozessen, die gemeinsam gestartet und gestoppt werden.



## Unit-Type **swap** - Swap-Dateien

- Swap-Units sind Swap-Dateien oder Swap-Partitionen, die von Systemd verwaltet werden.

# Systemd-Unit-Files

TODO

# Links

- <https://de.wikipedia.org/wiki/Systemd>
- <https://www.freedesktop.org/wiki/Software/systemd/>
- <https://www.digitalocean.com/community/tutorials/understanding-systemd-units-and-unit-files>
- <https://wiki.archlinux.org/title/Systemd>
- <https://www.tecmint.com/systemd-replaces-init-in-linux/>
- <https://www.linux-community.de/ausgaben/linuxuser/2014/04/systemd-als-schaltzentrale-fuer-das-linux-system/>