

systemd

- [bootup — System bootup process](#)

HowTo

start / stop

Rerun all generators, reload all unit files and recreate dependency tree:

```
systemctl daemon-reload
```

```
systemctl start servicename
```

info/status

```
# systemctl get-default  
multi-user.target
```

```
systemd-analyze blame  
systemd-analyze critical-chain  
systemd-analyze critical-chain network.service  
systemctl list-dependencies network.service  
systemctl --failed
```

```
systemd-cgtop
```

emergency mode

Prevent services from running

If services are defined with **DefaultDependencies=no**, they will be started unconditionally even in emergency mode, which can be wrong.

If mount fails

local-fs.target is configured to go to **OnFailure=emergency.target**. File `/etc/fstab` is parsed by **systemd-fstab-generator** to generate `.mount` units, which are **Required**.

To let mounts fail without switching to emergency mode, special attribute are introduced to `fstab`.
<http://www.freedesktop.org/software/systemd/man/systemd.mount.html>

Debug

Kernel commandline:

```
systemd.unit=emergency
```

```
systemd.log_level=debug systemd.log_target=console console=ttyS0,38400
```

Enable in u-boot:

```
setenv bootargs ${bootargs} systemd.unit=emergency.target
```

Enable debug console on tty9

```
systemctl enable debug-shell.service
```

Start service when MMC card is inserted

Create directory “dev-mmcblk0.device.wants” and put symlink to service inside it. List of devices:

```
systemctl --all --full -t device
```

Configuration

DefaultDependencies

Value for unit **DefaultDependencies** is set to **yes** by default. it will add following dependencies:

- **Conflicts=shutdown.target** to stop service during shutdown
- **Conflicts=umount.target** to umount mounts during shutdown
- **Requires=basic.target** and **After=basic.target**

Unit Types

- .service plain service - for running daemons
- .socket socket listeners - like inetd
- .device
- .path filesystem triggers (when file is changed)
- .mount
- .automount
- .swap
- .target
- .timer
- .snapshot
- .slice
- .scope

[Unit]

- Requires - units listed here, are also activated with unit. If any start fail, this unit will be not activated.
- Conflicts - negative dependencies, opposite to Requires
- Requisite - similar to require, but if not met, service will be not started (fail immediately). Usefull to perform action when removable device is inserted.
- Requisite=!local-fs.target
- BindsTo - similar to require, but also declares to stop unit when other units dissapers (aslo nice for removable devices handling)
- PartOf - stops/starts unit when unit listed is stopped/started
- After - start service after other services are started completely.
- Before

[Service]

- Type
 - Type=simple - default -
 - Type=forking - process will spawn child and exit. Child should live as daemon.
 - PIDFile=/var/run/service.pid - set it to give systemd possibility to known forked child PID
 - Type=oneshot - process should exit. Systemd is not waiting for finish. Like batch file.
 - usefull for setup some things, not starting daemons. See ReamainAfterExit=Yes)
 - Multiple ExecStart= allowed, and will be executed sequentially
 - Type=dbus - systemd assume process started when process name connects to D-Bus
 - Type=notify - systemd assume process started when process explicitly notify systemd using sd_notify()
 - Type=idle
- ExecStart= - first argument MUST be full path to executable
 - ExecStart=-/bin/false - minus sign = ignore exit code and assume always success
 - Multiple ExecStart= are allowed, and executed sequentially. If one command fails, other lines are not executed and unit is failed.
- ExecStartPre= - ExecStart= will run only if all ExecStartPre= command exits with success. May not to start long processes.
- ExecStartPost= - will be executed only if service has started successfully
- ExecStop=
- ExecReload=
- Nice=-20
- IOSchedulingClass=0
- IOSchedulingPriority=0
- StandardOutput=tty | none | journal+console
- Restart=always
- RestartSec=10s
- RemainAfterExit=Yes - keep service active status, even all process exits. usefull for

Type=oneshot

tricks

Condition doesn't skip service, but only service is not started but all requirements (after,before,wants, conflicts) works. So Service is not started but it is still disabling other services by Conflicts definition. If service is not started by Condition it is still conflicting

Remove startup console clearing

[/etc/systemd/system/getty@tty1.service.d/noclear.conf](#)

```
[Service]
TTYVTDIsallocate=no
```

udev

In udev rules.d add TAG+="systemd":

```
SUBSYSTEM=="input", KERNEL=="event[0-9]*", ENV{ID_INPUT_TOUCHSCREEN}=="1",
ATTRS{name}=="edt-ft53*", \
    SYMLINK+="input/lcd_sandwich_touchscreen input/touchscreen_edt", \
    TAG+="systemd", \
    OPTIONS+="link_priority=100"
```

And now device is visible from systemd:

```
systemctl -l -a
dev-input-event0.device
loaded active plugged /dev/input/event0
dev-input-lcd_sandwich_touchscreen.device
loaded active plugged /dev/input/lcd_sandwich_touchscreen
```

You can create refer to device, e.g. create .wants directory

```
/etc/systemd/system/ev-input-lcd_sandwich_touchscreen.device.wants
```

udev: start service

To start service if device is inserted:

[mu.rule](#)

```
..., TAG+="systemd", ENV{SYSTEMD_WANTS}="netctl-auto@mywifi.service"
```

do not mount crypted volumes

After switching to systemd by command

```
apt-get install systemd-sysv
```

, new problem occurs: system boot stops and waits for password for encrypted partition (with user data - no needed to system start).

To prevent automount of crypted volumes, additional parameters 'noauto' and 'noearly' must be added to /etc/crypttab:

[/etc/crypttab](#)

```
/MyMountPoint      /dev/vg_MyName/lv_MyName    none
luks,noauto,noearly
```

Debian: Set network service timeout

System startup can hang forever if there is no link on ethernet cable:

```
[    **] A start job is running for LSB: Raise network interf...38s / no
limit)
```

To set timeout, add file:

[/etc/systemd/system/networking.service.d/reduce-timeout.conf](#)

```
[Service]
TimeoutStartSec=15
```

Issues

init.d services not started

```
systemd-sysv-generator[320]: stat() failed on /etc/init.d/traccar, ignoring:
No such file or directory
systemd-sysv-generator[320]: stat() failed on /etc/init.d/ts3server,
```

```
ignoring: No such file or directory
systemd-sysv-generator[320]: stat() failed on /etc/init.d/mio_server,
ignoring: No such file or directory
```

Reason is that some init scripts are symlinked to files located in **/opt** directory, which is a different FS (in this case BTRFS subvolume), and it was not mounted when **systemd-sysv-generator** was started.

Solution: [How to mount non-root filesystem before systemd-sysv-generator runs?](#)

Under systemd-enabled systems, there's a new-with-systemd mount-option you can place in `/etc/fstab` — **x-initrd.mount**.

You also need to make sure that your filesystem's `fs_passno` is set to "0" ...and if your filesystem lives on an LVM2 volume, you need to update your GRUB2 config to ensure that the LVM gets onlined prior to systemd invoking the `systemd-sysv-generator` utility

From:

<https://niziak.spox.org/wiki/> - **niziak.spox.org**

Permanent link:

<https://niziak.spox.org/wiki/linux:systemd>

Last update: **2024/02/09 19:24**

