

# *Kurz erklärt*

---

systemd als Timer einsetzen

---

Günther Zander  
29. Dezember 2020

[ ● ◀ ] **systemd**

## [ana]cron vs. systemd

Das Programm *cron* ist ein Unix-Urgestein und dementsprechend auch gut getestet. Aber dieses Programm wurde für Host-Systeme entwickelt und war nicht dafür gedacht auf Personal Computer, Laptops oder RaspberryPi eingesetzt zu werden, da zur Zeit der Entwicklung diese Hardware noch nicht absehbar war.

Aus diesem Grund fehlte eine elementare Einstellung in dem Programm. Nämlich das eine Aufgabe gestartet werden kann in Abhängigkeit von dem Startzeitpunkt des Computers. Um diesen Mangel zu beheben wurde das Programm *anacron* entwickelt welches den Distributionen beiliegt.

In der Zwischenzeit wurde der Startprozess von GNU/Linux auf *systemd* umgestellt und dieses bietet die selben Möglichkeiten wie *cron* und *anacron*. Dieser Teilbereich von *systemd* wird Ihnen in diesem kurzem Artikel näher gebracht werden.

Diese geschieht am Beispiel einer zeitgesteuerten Indexierung der Datenbank von *recoll* welches im Artikel „Volltextsuche in Dateien unter GNU/Linux“ vorgestellt worden ist. Am Ende des Artikel finden Sie ein Installationsprogramm für diesen Fall und als Vorlage für ihre eigenen Projekte.

### Warum systemd?

Das Einrichten mittels *systemd* ist mit mehr Aufwand verbunden als eine Zeile in der [ana]crontab, die für die periodische Ausführung erforderlich wäre. Da [ana]cron in Zukunft immer mehr in den Hintergrund treten wird, ist es notwendig sich mit dem neuen System zu beschäftigen. Ein Vorteile von *systemd* ist, dass *systemd*-Jobs exakter gesteuert werden können (eigenes Environment, *cgroups*-Regeln, ...) und diese im *systemd*-Journal protokolliert werden.

Einige Nachteile sind im Moment noch, dass bei einem Fehler nicht automatisch eine E-Mail versendet wird und es keine zufällige Verzögerung gibt, um die gleichzeitige Ausführung zu vieler *systemd*-Jobs zu verhindern.

---

# Grundlagen

`systemctl list-timers` [-all]

Die zeitgesteuerte Ausführung von Programmen wird in der `systemd`-Umgebung durch einzelne Timer-Dateien gesteuert. Diese haben die Endung `.timer`. Einen Überblick über alle aktive Timer bekommen sie durch den Befehl `systemctl list-timers` und mit dem Zusatz `-all`, auch über die inaktiven Prozesse.

## Die `.timer`-Datei

Die `.timer`-Dateien liegen als einfache Textdateien vor. Die folgenden Zeilen zeigen die Datei `recoll.timer`, die für Aktualisierung einer Recoll-Datenbank verantwortlich ist, um die Volltextsuche immer parat zu haben. Diese Aktualisierung findet aber erst eine Stunde nach dem Start des Computers statt.

```
# /usr/lib/systemd/system/recoll.timer
[Unit]
Description=recoll timer
[Timer]
OnBootSec=1h
[Install]
WantedBy=basic.target
```

`OnBootSec=1h` bewirkt, dass das Prozess eine Stunde nach dem Start des Computers ausgeführt wird. Um einen Überblick über die Zeitangaben und Möglichkeiten zu erhalten sollten sie das Manual mit `man systemd.time` lesen.

## Die `.service`-Datei

Jede `.timer`-Datei hat eine `.service`-Datei, mit den Details zum auszuführenden Prozess.

```
# /usr/lib/systemd/system/recoll.service
[Unit]
Description=recoll automatic
[Service]
ExecStart=/usr/bin/recollindex
```

## Aktivierung des neuen Timers

Neu eingerichtete `.timer`-Dateien müssen sie mit `systemctl start` erstmalig startet und mit `systemctl enable` dauerhaft aktivieren. Damit Änderungen in bereits aktiven Timern aktiviert werden, führen Sie am einfachsten den folgenden Befehl aus, womit die geänderten Daten übernommen werden.

```
systemctl reenable --now recoll.timer
```

Nach dem Einrichten neuer `.timer`-Dateien sollten Sie immer mit `systemctl list-timers` sich vergewissern, dass `systemd` Ihre `.timer`-Datei auch korrekt eingebunden hat.

---

# Installationsprogramm

```
#!/bin/bash

# Install Recoll
# -----
# Author: Günther Zander, Hamburg 12/2020
# Licence: GPL
# File:    install-systemd-recoll.sh
# Version: 0.1
#
# Check root-Right
# -----
if [ "`id -gn`" != "root" ]; then
    echo; echo "Script bitte mit root-Rechten starten."; echo; exit 1
fi

# Create timer-File
# -----
cat >/usr/lib/systemd/system/recoll.timer <<EOF
# /usr/lib/systemd/system/recoll.timer
[Unit]
Description=recoll timer
[Timer]
OnBootSec=1h
[Install]
WantedBy=basic.target
EOF

# Create service-File
# -----
cat >/usr/lib/systemd/system/recoll.service <<EOF
# /usr/lib/systemd/system/recoll.service
[Unit]
Description=recoll automatic
[Service]
ExecStart=/usr/bin/recollindex
EOF

# Start .timer
# -----
systemctl start recoll.timer
systemctl enable recoll.timer

# Control Service
# -----
VAR=$(systemctl list-timers --all | grep -e ago -e left | grep recoll.service)
if [ -n "$VAR" ]
then
```

```
echo
echo -en "\e[0;32mOK \e[0m";
echo "Der Service von Recoll ist installiert und läuft."
echo
else
echo
echo -en "\e[1;31mFehler \e[0m";
echo "Es ist ein Fehler beim Start des Services von Recoll aufgetreten."
echo
systemctl list-timers --all | grep NEXT
systemctl list-timers --all | grep recoll.service
echo
fi
```