Linux - Watchdog

> Anleitung zur Installation und Konfiguration des Pakets **watchdog** (Link zur Projektseite).

> Die Anleitungen zum Aktivieren der Watchdogs für **Proxmox** und **Raspberry Pi** finden sich hier:
>
> Virtuelles Watchdog-Modul für Proxmox
>
> Hardware-Watchdog für Raspberry Pi

Installieren des Watchdog Pakets

```
apt install watchdog
```

Konfigurieren des Watchdogs

Folgende Änderungen in der Datei **/etc/watchdog** durchführen:

```
watchdog-device         = /dev/watchdog
watchdog-timeout        = 15
```

Testen des Watchdogs

> Der folgende Befehl triggert eine Kernel Panic worauf der Rechner neu bootet.

```
echo c > /proc/sysrq-trigger
```

Beispiele

> Folgende Beispiele lassen den Watchdog einen **reboot** durchführen **wenn die Bedingung NICHT zutrifft**.

Interface überträgt Daten:

```
interface               = eth0
```

IP-Adresse pingbar:

```
ping                    = 172.16.0.1
```

Max. Load ist innerhalb 1 Minute unter 24%, innerhalb 5 Minuten unter 18%, innerhalb 15 Minuten unter 12%:

```
max-load-1              = 24
max-load-5              = 18
max-load-15             = 12
```

Skripte

Skripte können in das Verzeichnis **/etc/watchdog.d** (Vorgabe in der **watchdog.conf**) gelegt werden und werden Automatisch vom Watchdog zum Testen verwendet. Der Watchdog wird bei einem **exit 0** den Watchdog-Timer zurücksetzen. Bei einem **exit 255** (-1) wird ein reboot durchgeführt.

```
#!/bin/bash

# Es wird ein Befehl ausgeführt und die Ausgabe in die Variable geschrieben
# Hier wird z.B. nach einen USB-Gerät gesucht, bei welchem in der Zeile "Titanium" vorkommt
listusb=`lsusb | grep Titanium`

if [[ -z  $listusb ]]; then # Wenn die Variable leer ist (Gerät nicht vorhanden)
        exit 255 # Rückgabewert -1 Neustart (reboot)
fi

exit 0 # Wert 0 setzt Watchdog Timer zurück
```

> Als Rückgabewerte (exit codes) für den Watchdog stehen folgende Werte zu Verfügung:
>
> 0 (exit 0): Kein Fehler - Watchdog wird zurückgesetzt
> -1 (exit 255): Neustart (reboot)
> -2 (exit 254): Zurücksetzen (reset)
> -3 (exit 253): Maximale durchschnittliche Auslastung überschritten.
> -4 (exit 252): Die Innentemperatur ist zu hoch.
> -5 (exit 251): /proc/loadavg enthält keine (oder nicht genügend) Daten.
> -6 (exit 250): Die angegebene Datei wurde im angegebenen Intervall nicht geändert.
> -7 (exit 249): /proc/meminfo enthält ungültige Daten.
> -8 (exit 248): Der Kindprozess wurde durch ein Signal beendet.
> -9 (exit 247): Der Kindprozess ist nicht rechtzeitig zurückgekehrt.
> -10 (exit 246): Für den persönlichen Gebrauch frei.

/etc/watchdog.conf (Beispieldatei)

```
# ======================================================================
# Configuration for the watchdog daemon. For more information on the
# parameters in this file use the command 'man watchdog.conf'
# ======================================================================

# ================== The hardware timer settings ====================
#
# For this daemon to be effective it really needs some hardware timer
# to back up any reboot actions. If you have a server then see if it
# has IPMI support. Otherwise for Intel-based machines try the iTCO_wdt
# module, otherwise (or if that fails) then see if any of the following
# module load and work:
#
# it87_wdt it8712f_wdt w83627hf_wdt w83877f_wdt w83977f_wdt
#
# If all else fails then 'softdog' is better than no timer at all!
# Or work your way through the modules listed under:
#
# /lib/modules/`uname -r`/kernel/drivers/watchdog/
#
# To see if they load, present /dev/watchdog, and are capable of
# resetting the system on time-out.

# Uncomment this to use the watchdog device driver access "file".

watchdog-device         = /dev/watchdog

# Uncomment and edit this line for hardware timeout values that differ
# from the default of one minute.

watchdog-timeout        = 15

# If your watchdog trips by itself when the first timeout interval
# elapses then try uncommenting the line below and changing the
# value to 'yes'.

#watchdog-refresh-use-settimeout        = auto
```

```
# If you have a buggy watchdog device (e.g. some IPMI implementations)
# try uncommenting this line and setting it to 'yes'.

#watchdog-refresh-ignore-errors = no

# ====================== Other system settings =======================
#
# Interval between tests. Should be a couple of seconds shorter than
# the hardware time-out value.

#interval               = 1

# The number of intervals skipped before a log message is written (i.e.
# a multiplier for 'interval' in terms of syslog messages)

#logtick        = 1

# Directory for log files (probably best not to change this)

#log-dir                = /var/log/watchdog

# Email address for sending the reboot reason. This needs sendmail to
# be installed and properly configured. Maybe you should just enable
# syslog forwarding instead?

#admin                  = root

# Lock the daemon in to memory as a real-time process. This greatly
# decreases the chance that watchdog won't be scheduled before your
# machine is really loaded.

realtime                = yes
priority                = 1

# ====================== How to handle errors  ======================
#
# If you have a custom binary/script to handle errors then uncomment
# this line and provide the path. For 'v1' test binary files they also
# handle error cases.
```

```
#repair-binary          = /usr/sbin/repair
#repair-timeout         = 60

# The retry-timeout and repair limit are used to handle errors in a
# more robust manner. Errors must persist for longer than this to
# action a repair or reboot, and if repair-maximum attempts are
# made without the test passing a reboot is initiated anyway.

#retry-timeout          = 60
#repair-maximum         = 1

# Configure the delay on reboot from sending SIGTERM to all processes
# and to following up with SIGKILL for any that are ignoring the polite
# request to stop.

#sigterm-delay          = 5

# ====================== User-specified tests ========================
#
# Specify the directory for auto-added 'v1' test programs (any executable
# found in the 'test-directory should be listed).

#test-directory = /etc/watchdog.d

# Specify any v0 custom tests here. Multiple lines are permitted, but
# having any 'v1' programs/scripts discovered in the 'test-directory' is
# the better way.

#test-binary            =

# Specify the time-out value for a test error to be reported.

#test-timeout           = 60

# ====================== Typical tests ================================
#
# Specify any IPv4 numeric addresses to be probed.
# NOTE: You should check you have permission to ping any machine before
# using it as a test. Also remember if the target goes down then this
```

```
# machine will reboot as a result!

#ping                   = 172.16.0.1
#ping                   = 192.168.1.1

# Set the number of ping attempts in each 'interval' of time. Default
# is 3 and it completes on the first successful ping.
# NOTE: Round-trip delay has to be less than 'interval' / 'ping-count'
# for test success, but this is unlikely to be exceeded except possibly
# on satellite links (very unlikely case!).

#ping-count             = 3

# Specify any network interface to be checked for activity.

#interface              = eth0

# Specify any files to be checked for presence, and if desired, checked
# that they have been updated more recently than 'change' seconds.

#file                   = /var/log/syslog
#change                 = 1407

# Uncomment to enable load average tests for 1, 5 and 15 minute
# averages. Setting one of these values to '0' disables it. These
# values will hopefully never reboot your machine during normal use
# (if your machine is really hung, the loadavg will go much higher
# than 25 in most cases).

#max-load-1             = 24
#max-load-5             = 18
#max-load-15            = 12

# Check available memory on the machine.
#
# The min-memory check is a passive test from reading the file
# /proc/meminfo and computed from MemFree + Buffers + Cached
# If this is below a few tens of MB you are likely to have problems.
#
# The allocatable-memory is an active test checking it can be paged
```

```
# in to use.
#
# Maximum swap should be based on normal use, probably a large part of
# available swap but paging 1GB of swap can take tens of seconds.
#
# NOTE: This is the number of pages, to get the real size, check how
# large the pagesize is on your machine (typically 4kB for x86 hardware).

#min-memory             = 1
#allocatable-memory     = 1
#max-swap = 0

# Check for over-temperature. Typically the temperature-sensor is a
# 'virtual file' under /sys and it contains the temperature in
# milli-Celsius. Usually these are generated by the 'sensors' package,
# but take care as device enumeration may not be fixed.

#temperature-sensor     =
#max-temperature        = 90

# Check for a running process/daemon by its PID file. For example,
# check if rsyslogd is still running by enabling the following line:

#pidfile                = /var/run/rsyslogd.pid
```

Revision #18
Created 2026-03-30 14:42:46 UTC by Admin
Updated 2026-03-30 18:36:04 UTC by Admin