

HOWTO zu UNIX-Signalen

(C) 2007-2017 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Open Source Training and Consulting GmbH
<http://www.ostc.de>

\$Id: unix-signals-HOWTO.txt,v 1.12 2019/11/26 19:37:07 tsbirn Exp \$

Dieses Dokument beschreibt die Signale von UNIX.

INHALTSVERZEICHNIS

- 1) Überblick
- 2) Tastendruck-Signale
- 3) Kommando "kill"
- 4) Weitere "kill"-Kommandos

1) Überblick

UNIX-Prozesse können sich über Signale miteinander unterhalten und bestimmte Verhaltensweisen auslösen. Signale werden erzeugt durch:

- * das Betriebssystem selbst (automatisch)
- * durch Prozesse (manuell)
- * durch das Kommando "kill" (manuell)

Eine vollständige Liste aller Signale + ihre Nummer erhält man durch "kill -l" (list) und eine Beschreibung durch "man 7 signal". Die wichtigsten Signale sind (die Nr ist leider nicht einheitlich über alle UNIX-Systeme hinweg):

| Name | Nr | Bedeutung |
|-----------|----|---|
| SIGEXIT | 0 | Bash: Am Skriptende (exit) ausgelöst |
| SIGDEBUG | ? | Bash: Nach jedem "einfachen Kmdo" ausgelöst |
| SIGRETURN | ? | Bash: Nach jedem Shell-Funktionsaufruf + ./source-Kmdo |
| SIGERR | ? | Bash: Nach jedem "einfachen Kmdo" mit Exit-Status != 0 a. |
| SIGHUP | 1 | Konfigurationsdatei erneut einlesen (Daemon) [hangup] |
| SIGINT | 2 | Abbrechen durch <Strg-C> [interrupt] |
| SIGKILL | 9 | Bedingungsloser Prozessabbruch [kill] |
| SIGTERM | 15 | Prozess beenden (Standard-Signal von "kill") [terminate] |
| SIGUSR1 | 10 | Benutzerspezifisch Nr 1 [user] |
| SIGUSR2 | 12 | Benutzerspezifisch Nr 2 [user] |
| SIGCONT | 18 | Ausgabe weiterlaufen lassen durch <Strg-Q> [continue] |
| SIGSTOP | 19 | Ausgabe anhalten durch <Strg-S> [stop] |
| SIGTSTP | 20 | In Hintergrund stellen durch <Strg-Z> [terminate stop] |
| SIGQUIT | 3 | Prozessende erreicht [quit] |
| SIGILL | 4 | Nicht erlaubte Anweisung gefunden [illegal] |
| SIGBUS | 7 | Bus-Zugriffsverletzung (Alignment) [bus] |
| SIGFPE | 8 | Mathematik-Fehler [floating point exception] |
| SIGSEGV | 11 | Fehlerhafter Speicherzugriff [segment violation] |
| SIGPIPE | 13 | Prozess in Pipeline nicht mehr da [broken pipe] |
| SIGALRM | 14 | Timeout hat stattgefunden [alarm] |
| SIGCHLD | 17 | Ein Kindprozess hat sich beendet [child] |

Alle Signale außer SIGKILL (9) und SIGTSTP (20) können von einem Prozess abgefangen und so z.B. ignoriert werden.

HINWEIS: Die mit "Bash" gekennzeichneten Signale sind nur innerhalb eines Bash-Skriptes relevant, es sind keine echten Signale des UNIX-Systems.

2) Tastendruck-Signale

Durch Drücken der Taste <Strg-C> wird das Signal SIGINTR (2) an den aktuell laufenden Prozess geschickt. Normalerweise wird es dadurch abgebrochen. Es gibt aber auch Programme, die dieses Signal ignorieren (z.B. die Shell).

<Strg-C> schickt das Signal SIGINTR (2) an den aktuell laufenden Prozess. Normalerweise wird es dadurch abgebrochen. Es gibt aber auch Programme, die dieses Signal ignorieren (z.B. die Shell).

<Strg-Z> schickt das Signal SIGTSTP (18) an den aktuell laufenden Prozess. Er wird dadurch angehalten und in den Hintergrund gestellt. Dieses Signal kann nicht abgefangen werden.

<Strg-S> schickt das Signal SIGSTOP (17) an den aktuell laufenden Prozess. Seine Ausgabe wird angehalten. <Strg-Q> schickt das Signal SIGCONT (19), das die Ausgabe des aktuellen Prozesses wieder aktiviert.

3) Kommando "kill"

Mit "kill" werden Signale manuell an Prozesse verschickt. Dazu ist erst ihre Prozess-Nummer z.B. mit "ps aux" (BSD) oder "ps -ef" (SYSV) zu ermitteln. Die Standardform schickt das Signal SIGTERM (15).

```
kill 1234...          # oder
kill -15 1234...     # oder
kill -TERM 1234      # oder
kill -SIGTERM 1234
```

Der Prozess hat dann noch Zeit zum Aufräumen, z.B. kann er Puffer rausschreiben, Dateien schließen und Zwischendateien löschen. Ein Prozess kann sich aber auch gegen dieses Signal wehren und sich weigern aufzuräumen.

Mit dem Signal SIGKILL (9) wird ein Prozess auf jeden Fall gezwungen, sich zu beenden. Allerdings kann er dann nichts mehr aufräumen, alles bleibt so stehen und liegen, wie es zum Zeitpunkt des Abbruchs war.

```
kill -9 1234...      # oder
kill -KILL           # oder
kill -SIGKILL
```

Will man einem Daemon (Disk and Execution Monitor), der ständig im Hintergrund läuft, mitteilen, daß sich seine Konfigurationsdatei geändert hat, so kann man ihm das Signal SIGHUP (1) schicken, damit er diese erneut einliest.

```
kill -1 1234...      # oder
kill -HUP 1234...    # oder
kill -SIGHUP 1234...
```

4) Weitere "kill"-Kommandos

Beenden ALLER Kommandos mit dem Namensbestandteil NAME:

```
killall NAME
pkill NAME
```

Suchen der PID aller Prozesse mit dem Namensbestandteil NAME:

```
pgrep NAME
```