

HOWTO zu UNIX-Signalen
 (C) 2007 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
 OSTC GmbH, <http://www.ostc.de>
 \$Id: unix-signals-HOWTO.txt,v 1.5 2009-10-06 19:23:15 tsbirn Exp \$

Dieses Dokument beschreibt die Signale von UNIX.

Inhaltsverzeichnis

- 1) Übersicht
- 2) Tastendruck-Signale
- 3) Kommando "kill"

1) Übersicht

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

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

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

Name	Nr	Bedeutung
SIGEXIT	0	"Ende des Skripts" in der Bash
SIGHUP	1	Konfigurationsdatei erneut einlesen (Daemon) [hangup]
SIGINT	2	Abbrechen durch <Strg-C> [interrupt]
SIGKILL	9	Bedingungsloser Prozessabbruch
SIGTERM	15	Prozess beenden (Standard-Signal von "kill") [terminate]
SIGCONT	18	Ausgabe weiterlaufen lassen durch <Strg-Q>
SIGSTOP	19	Ausgabe anhalten durch <Strg-S>
SIGTSTP	20	In Hintergrund stellen durch <Strg-Z>
SIGQUIT	3	Prozessende erreicht
SIGILL	4	Nicht erlaubte Anweisung gefunden [illegal]
SIGBUS	7	Bus-Zugriffsverletzung (Alignment)
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.

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).

Durch Drücken der Taste <Strg-Z> wird das Signal SIGTSTP (20) an den aktuell laufenden Prozess geschickt. Er wird dadurch angehalten und in den Hintergrund gestellt. Dieses Signal kann nicht abgefangen werden.

Durch Drücken der Taste <Strg-S> (Signal SIGSTOP = 19) wird die Ausgabe des aktuellen Prozesses angehalten, durch <Strg-Q> (Signal SIGCINT = 18) wird 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 aufzuhören.

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...
```