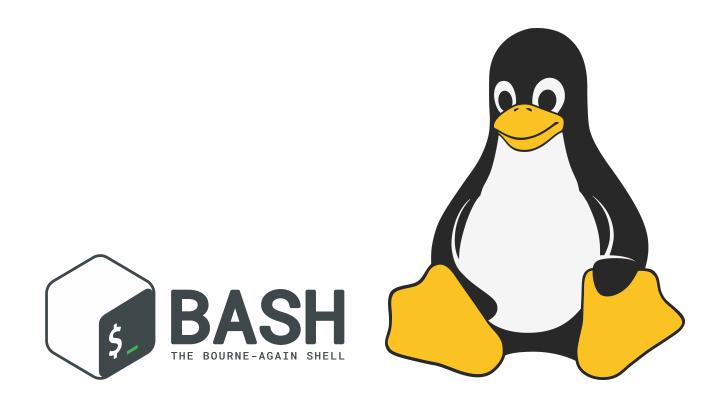
kill - Signale senden



Inhaltsverzeichnis

- Signale senden mit kill
- <u>Die wichtigsten Signale</u>
- Wie beendet man einen Prozess?
- Key Takeaways Signale

Signale senden mit kill

Mit dem Kommando kill können Signale an Prozesse gesendet werden. Der Prozess, der das Signal empfängt, kann auf unterschiedliche Weise darauf reagieren:

- Der Prozess kann das Signal ignorieren.
- Der Prozess kann das Signal abfangen und darauf reagieren.
- Der Prozess fängt das Signal nicht ab und wird beendet.

Von den 64 Signalen, die in einem modernen Linux-System zur Verfügung stehen, sind die folgenden die wichtigsten:

Die wichtigsten Signale

- SIGINT (Signal 2): Interrupt-Signal, das durch die Tastenkombination Ctrl-C vom Terminal-Benutzer erzeugt wird. Standardaktion: Beenden des Prozesses. Es kann auch abgefangen oder ignoriert werden.
- SIGTERM (Signal 15): Terminierungs-Signal, das durch kill Kommando erzeugt wird. Standardaktion: Beenden des Prozesses. Das Signal kann auch abgefangen oder ignoriert werden.
- **SIGKILL** (**Signal 9**): Kill-Signal, das durch kill -9 -Kommando erzeugt wird. Standardaktion: Sofortiges Beenden des Prozesses. !!! Dieses Signal kann nicht abgefangen oder ignoriert werden !!!

Wie beendet man einen Prozess?

Signal 2 (SIGINT) senden:

Blockiert ein Vordergrund-Prozess das Terminal (z. B. weil er in einer Endlosschleife hängt), kann man ihn meist mit der Tastenkombination Ctrl-C beenden. Ctrl-C sendet das Signal 2 an den Prozess und bricht ihn damit in der Regel ab. Hintergrundprozesse sind mit Ctrl-C nicht erreichbar. Man kann ihnen jedoch das Signal 2 mit dem kill-Kommando senden: kill-2 <pid> oder kill-INT <pid>. Signal 2 mit dem kill -Kommando zu senden, ist eher unüblich. Meist nimmt man mit dem kill -Kommando die Signale 15 oder 9.

Signal 15 (SIGTERM) senden:

Wenn Ctrl-C bzw. das Senden von Signal 2 nicht funktioniert (weil dieses Signal ignoriert wird), dann kann man sich an einem anderen Terminal anmelden und den Prozess mit kill -15 oder kill -TERM beenden. Der Prozess reagiert darauf normalerweise mit einer geregelten Beendigung. (Er speichert ggf. seine Daten und gibt alle belegten Ressourcen frei, bevor er sich beendet.) Der Prozess kann aber auch so programmiert sein, dass er Signal 15 (SIGTERM) ignoriert. Dann ist die Beendigung auf diesem Weg nicht möglich.

Signal 9 (SIGKILL) senden:

• Wenn auch kill -15 nicht funktioniert, dann bleibt als letzter Ausweg kill -9, bzw. kill -KILL. Dieses Signal kann nicht ignoriert oder abgefangen werden. Der Prozess wird sofort beendet, ohne dass er die Möglichkeit hat, sich kontrolliert zu beenden oder Ressourcen freizugeben. Dies kann zu Datenverlust führen.

Hintergrundprozesse sind mit Ctrl-C nicht erreichbar. Sie können nur mit dem kill -Kommando beendet werden.

Die Signalnummer 15 ist beim kill -Kommando der Standardwert. Sie kann weggelassen werden.

Senden von Signal 2 (SIGINT) mit Ctrl+C oder kill -2

```
hermann@debian:~$ tty
/dev/pts/1
hermann@debian:~$ sleep 60 # fg process
^C
hermann@debian:~$ # fg process was interrupted with Ctr1-C
hermann@debian:~$ sleep 60 & # bg process
[1] 37617
hermann@debian:~$ kill -2 37617 # send SIGINT to bg process
hermann@debian:~$
[1]+ Unterbrechung sleep 60
```

Statt kill -2 <pid> kann auch die symboische Signalbezeichnung kill -INT <pid> verwendet werden.

Senden von Signal 15 (SIGTERM) mit kill -15

```
hermann@debian:~$ sleep 60 & # bg process
[1] 37620
hermann@debian:~$ kill -15 37620
hermann@debian:~$
[1]+ Beendet sleep 60
```

Statt kill -15 <pid> kann auch kill -TERM <pid> verwendet werden. Man kann die Signalnummer 15 auch weglassen, da sie der Standard-Signalwert ist.

Senden von Signal 9 (SIGKILL) mit kill -9

```
hermann@debian:~$ sleep 60 & # bg process
[1] 37626
hermann@debian:~$ kill -9 37626
hermann@debian:~$
[1]+ Getötet sleep 60
```

Statt kill -9 <pid> kann auch kill -KILL <pid> verwendet werden.

Das Signal 9 kann keinesfalls ignoriert oder abgefangen werden.

Key Takeaways - Signale

- Wenn sich ein Vordergrund-Prozess im Terminal (TTY)
 "festgefahren" hat, dann versuchen Sie ihn zuerst mit Ctrl-C zu beenden.
- Wenn das nicht funktioniert, oder wenn es sich um einen Hintergrund-Prozess handelt oder um einen Prozess, der nicht an ein TTY gebunden ist, dann versuchen Sie es mit kill -15 <pid>.
 Damit geben Sie dem Prozess die Chance, sich kontrolliert zu beenden. (Er kann dann z.B. noch Daten speichern oder belegte Ressourcen freigeben. Er könnte aber auch so programmiert sein, dass er das Signal 15 ignoriert.)