NAME

timeout - run a command with a time limit

SYNOPSIS

timeout [**--signal** *sig* | **-s** *sig*] [**--preserve-status**] [**--kill-after** *time* | **-k** *time*] [**--foreground**] *duration* *command* [*args* ...]

DESCRIPTION

timeout starts the *command* with its *args*. If the *command* is still running after *duration*, it is killed. By default, SIGTERM is sent. The special *duration*, zero, signifies no limit. Therefore a signal is never sent if *duration* is 0.

The options are as follows:

--preserve-status
Exit with the same status as *command*, even if it times out and is killed.

--foreground
Do not propagate timeout to the children of *command*.

-s *sig*, **--signal** *sig*
Specify the signal to send on timeout. By default, SIGTERM is sent.

-k *time*, **--kill-after** *time*
Send a SIGKILL signal if *command* is still running after *time* after the first signal was sent.

DURATION FORMAT

duration and *time* are non-negative integer or real (decimal) numbers, with an optional unit-specifying suffix. Values without an explicit unit are interpreted as seconds.

Supported unit symbols are:

s seconds
m minutes
h hours
d days

EXIT STATUS

If the timeout was not reached, the exit status of *command* is returned.

If the timeout was reached and **--preserve-status** is set, the exit status of *command* is returned. If

--preserve-status is not set, an exit status of 124 is returned.

If *command* exits after receiving a signal, the exit status returned is the signal number plus 128.

If *command* refers to a non-existing program, the exit status returned is 127.

If *command* is an otherwise invalid program, the exit status returned is 126.

If an invalid parameter is passed to **-s** or **-k**, the exit status returned is 125.

EXAMPLES

Run sleep(1) with a time limit of 4 seconds. Since the command completes in 2 seconds, the exit status is 0:

```
$ timeout 4 sleep 2
$ echo $?
0
```

Run sleep(1) for 4 seconds and terminate process after 2 seconds. 124 is returned since no **--preserve-status** is used:

```
$ timeout 2 sleep 4
$ echo $?
124
```

Same as above but preserving status. Exit status is 128 + signal number (15 for *SIGTERM*):

```
$ timeout --preserve-status 2 sleep 4
$ echo $?
143
```

Same as above but sending *SIGALRM* (signal number 14) instead of *SIGTERM*:

```
$ timeout --preserve-status -s SIGALRM 2 sleep 4
$ echo $?
142
```

Try to fetch(1) the PDF version of the FreeBSD Handbook. Send a *SIGTERM* signal after 1 minute and send a *SIGKILL* signal 5 seconds later if the process refuses to stop:

```
$ timeout -k 5s 1m fetch \
> https://download.freebsd.org/ftp/doc/en/books/handbook/book.pdf
```

SEE ALSO

kill(1), nohup(1), signal(3), daemon(8)

STANDARDS

The **timeout** utility is compliant with the IEEE Std 1003.1-2024 ("POSIX.1") specification.

HISTORY

The **timeout** command first appeared in FreeBSD 10.3.

AUTHORS

Baptiste Daroussin <*bapt@FreeBSD.org*> and
Vsevolod Stakhov <*vsevolod@FreeBSD.org*>