Table Of Contents

nixCraft: Linux Tips, Hacks, Tutorials, And Ideas In Blog Format
http://www.cyberciti.biz/

Home > Faq > BASH Shell

Alarm clock: How To Set Timeout For A Shell Command

Posted by Vivek Gite <vivek@nixcraft.com>

[1]

Q. How can I run a command called foo, and have it timeout / abort after 10 seconds under GNU/Linux running bash shell or script? How do I run command under an alarm clock?

A. You should able to use python / ruby / php or perl to set such timer. Shell do not have in built facility for timeout a command.

[2]

Perl program

Here is sample perl code:

```
perl -e 'alarm shift @ARGV; exec @ARGV' 5 foo arg1 arg2
```

You may define it as shell function:

```
alarm() { perl -e 'alarm shift; exec @ARGV' "$@"; }
```

And call it as follows (wait 20 seconds before alrming foo command:)

```
alarm 20 foo arg1
```

Sample shell script

You can use following shell script to set timeout for a command:

```
:
##########################################################################
# Shellscript: timeout - set timeout for a command
# Author     : Heiner Steven <heiner.steven@odn.de>
# Date       : 29.07.1999
# Category   : File Utilities
# Requires   :
# SCCS-Id.   : @(#) timeout 1.3 03/03/18
##########################################################################
# Description
#    o Runs a command, and terminates it (by sending a signal) after
# a specified time period
#    o This command first starts itself as a "watchdog" process in the
# background, and then runs the specified command.
# If the command did not terminate after the specified
# number of seconds, the "watchdog" process will terminate
# the command by sending a signal.
#
# Notes
#    o Uses the internal command line argument "-p" to specify the
# PID of the process to terminate after the timeout to the
# "watchdog" process.
#    o The "watchdog" process is invoked by the name "$0", so
# "$0" must be a valid path to the script.
#    o If this script runs in the environment of the login shell
# (i.e. it was invoked using ". timeout command...") it will
# terminate the login session.
##########################################################################

PN=`basename "$0"`    # Program name
```

```
VER='1.3'

TIMEOUT=5     # Default [seconds]

Usage () {
    echo >&2 "$PN - set timeout for a command, $VER
usage: $PN [-t timeout] command [argument ...]
     -t: timeout (in seconds, default is $TIMEOUT)"
    exit 1
}

Msg () {
    for MsgLine
    do echo "$PN: $MsgLine" >&2
    done
}

Fatal () { Msg "$@"; exit 1; }

while [ $# -gt 0 ]
do
    case "$1" in
 -p) ParentPID=$2; shift;; # Used internally!
 -t) Timeout="$2"; shift;;
 --) shift; break;;
 -h) Usage;;
 -*) Usage;;
 *) break;;    # First file name
    esac
    shift
done

: ${Timeout:=$TIMEOUT}    # Set default [seconds]

if [ -z "$ParentPID" ]
then
    # This is the first invokation of this script.
    # Start "watchdog" process, and then run the command.
    [ $# -lt 1 ] && Fatal "please specify a command to execute"
    "$0" -p $$ -t $Timeout &  # Start watchdog
    #echo >&2 "DEBUG: process id is $$"
    exec "$@"     # Run command
    exit 2     # NOT REACHED
else
    # We run in "watchdog" mode, $ParentPID contains the PID
    # of the process we should terminate after $Timeout seconds.
    [ $# -ne 0 ] && Fatal "please do not use -p option interactively"

    #echo >&2 "DEBUG: $$: parent PID to terminate is $ParentPID"

    exec >/dev/null 0<&1 2>&1 # Suppress error messages
    sleep $Timeout
    kill $ParentPID &&    # Give process time to terminate
     (sleep 2; kill -1 $ParentPID) &&
 (sleep 2; kill -9 $ParentPID)
    exit 0
fi
```

(Credit: Heiner Steven [3])

doalarm c program

Download doalarm program:

```
$ wget http://pilcrow.madison.wi.us/sw/doalarm-0.1.7.tgz
```

Untar program:

```
$ tar -zxvf doalarm-0.1.7.tg
```

Compile doalarm:

```
$ cd doalarm-0.1.7
$ make
$ ./doalarm
```

Sample output:

```
Error: missing required parameter
Usage: doalarm [-hr] [-t type] sec command [arg...]
Run command under an alarm clock.

Options:
 -t          Type of timer: 'real' (SIGALRM), 'virtual' (SIGVTALRM),
    --timer= 'profile' (SIGPROF), 'cpu' (SIGXCPU).  Default 'real'.
 -r --recur        Recurring alarm, every sec seconds.
 -h --help         Show help text (this message).
    --version      Display version.

doalarm 0.1.7 (14 Dec 2001)
```

Run foo as follows:

```
$ doalarm 20 foo
```

URLs in this post:

[1] Image: **http://www.cyberciti.biz/faq/category/bash-shell/**
[2] Image: **http://www.cyberciti.biz/faq/category/perl/**
[3] Heiner Steven: **http://www.shelldorado.com/scripts/cmds/timeout**