**Linux / UNIX: Bash Script Sleep or Delay a Specified Amount of Time**

How to pause bash shell script on a Linux or Unix-like systems?

You can use the sleep command to add a delay for a specified amount of time. The syntax is as follows for gnu/bash sleep command:

sleep NUMBER[SUFFIX]

Where SUFFIX may be:

s for seconds (the default)

m for minutes

h for hours

d for days

Please note that the sleep command in BSD family of operating systems (such as FreeBSD) or macOS/mac OS X does NOT take any suffix arguments (m/h/d). It only takes arguments in seconds. So syntax for sleep command for Unix like system is:

sleep NUMBER

**Examples**

To sleep for 5 seconds:

sleep 5

Want to sleep for 2 minutes:

sleep 2m

Halt or sleep for 3 hours, use:

sleep 3h

To sleep for 5 days:

sleep 5d

BSD/Unix/macOS implementations of sleep command have required that number be an integer, and only accepted a single argument without a suffix.

However, GNU/Linux version of sleep command accepts arbitrary floating point numbers.

To sleep for 1.5 seconds:

sleep 1.5

To sleep for .5 seconds:

sleep .5

The most common usage is:

run commmand1, sleep for 1 minute and finally run command2

command1 && sleep 1m && command2

sleep in bash for loop

for i in {1..10}

do

do\_something\_here

sleep 5s

done

run while loop to display date and hostname on screen

while [ : ]

do

clear

tput cup 5 5

date

tput cup 6 5

echo "Hostname : $(hostname)"

sleep 1

done

sleep Command Bash Script Example

Here is a simple example:

#!/bin/bash

echo "Hi, I'm sleeping for 5 seconds..."

sleep 5

echo "all Done."

In this example, create the lock directory. I’m using [/\\:.-] here to ensure that we don’t use the same name that we are using for the .o file. Also, base the name on the expected object file name, since that is what matters with a parallel build.

lockdir=`echo "$cofile" | sed -e 's|[/\\:.-]|\_|g'`.d

while true; do

if mkdir "$lockdir" >/dev/null 2>&1; then

break

fi

sleep 1

done

You can pause a bash shell script for 5 second before continuing.

Use the read command:

read -p "text" -t 5

read -p "Waiting five secs for Cloudflare to clear cache...." -t 5

echo "Generating pdf file now ...."

Sample outputs:

Waiting five secs for Cloudflare to clear cache....

Generating pdf file now ....

Where,

-p "text" : Show the text without a trailing newline before time out.

-t N : Set time out to 5 seconds.