

Ask questions and share your thoughts on the future of Stack Overflow. Join our first live community AMA this Wednesday, February 26th, at 3 PM ET. [Learn more](#)



# bash script to change directory and execute command with arguments

Asked 13 years, 2 months ago   Modified 9 years, 10 months ago   Viewed 49k times



12

I am trying to do the following task: write a shell script called `changedir` which takes a directory name, a command name and (optionally) some additional arguments. The script will then change into the directory indicated, and executes the command indicated with the arguments provided.



Here an example:



```
$ sh changedir /etc ls -al
```



This should change into the `/etc` directory and run the command `ls -al`.

So far I have:

```
#!/bin/sh
directory=$1; shift
command=$1; shift
args=$1; shift
cd $directory
$command
```

If I run the above like `sh changedir /etc ls` it changes and lists the directory. But if I add arguments to the `ls` it does not work. What do I need to do to correct it?

bash

Share   Improve this question   Follow

edited Dec 3, 2011 at 18:58



Lesmana

27.1k ● 12 ● 83 ● 87

asked Dec 3, 2011 at 17:16



frodo

1,063 ● 6 ● 17 ● 33

## 2 Answers

Sorted by: Highest score (default)



You seemed to be ignoring the remainder of the arguments to your command.

If I understand correctly you need to do something like this:

20



```
#!/bin/sh
cd "$1"      # change to directory specified by arg 1
shift       # drop arg 1
cmd="$1"     # grab command from next argument
shift       # drop next argument
"$cmd" "$@"  # expand remaining arguments, retaining original word
separations
```



A simpler and safer variant would be:

```
#!/bin/sh
cd "$1" && shift && "$@"
```

Share Improve this answer Follow

edited Apr 28, 2015 at 6:43

answered Dec 3, 2011 at 17:20



**CB Bailey**

793k ● 107 ● 644 ● 665

Thank you Charles ....am I right in making the last line above to be "\$cmd" "\$@" to make the "arguments" optional to the command i.e. if I run "./changedir /etc/ ls" it will work ok and also " ./indir /etc/ ls -al" will also work ok – [frodo](#) Dec 3, 2011 at 17:41

@frodo: Yes, you are right. "\$@" is better because it preserves arguments as they are passed. It's what I meant but not what I wrote! – [CB Bailey](#) Dec 3, 2011 at 18:24



-1



Since there can probably be more than a single argument to a command, i would recommend using quotation marks. Something like this:

```
sh changedir.sh /etc "ls -lsah"
```

Your code would be much more readable if you ommited the 'shift':

```
directory=$1;
command=$2;
cd $directory
$command
```

or simply

```
cd DIRECTORY_HERE; COMMAND_WITH_ARGS_HERE
```

Share Improve this answer Follow

edited Dec 5, 2011 at 9:25

answered Dec 3, 2011 at 17:40



**dgasper**

212 ● 2 ● 7

- 
- 1 [This way lies madness](#). @Charles Bailey's method handles arguments much more cleanly.  
– [Gordon Davisson](#) Dec 3, 2011 at 23:09
- 

True that handling more complex commands could be cumbersome but for the simple case mentioned here it should suffice. It is also much clearer what the directory part and the command parts are, since the arguments really belong to the command and have no meaning on their own. – [dgasper](#) Dec 5, 2011 at 9:20

---