

How do I run a program with a different working directory from current, from Linux shell?

Using a Linux shell, how do I start a program with a different working directory from the current working directory?

308

For example, I have a binary file `helloworld` that creates the file `hello-world.txt` in the current directory. This file is inside of directory `/a` . Currently I am in directory `/b` . I want to start my program running `../a/helloworld` and get the `hello-world.txt` somewhere in a third directory `/c` .



86

linux

shell

environment

edited May 2 '15 at 23:29



Mark Amery

68.7k

32

266

315

asked Apr 24 '09 at 15:36



Anton Daneyko

2,467

4

23

43

5 I discovered the hard way that `su` resets the working directory to the home directory of user you specify before running any `-c` commands. This was very helpful to me. – Patrick M Jan 16 '14 at 19:31

11 Answers



Call the program like this:

500

```
(cd /c; /a/helloworld)
```



The parentheses cause a sub-shell to be spawned. This sub-shell then changes its working directory to `/c` , then executes `helloworld` from `/a` . After the program exits, the sub-shell terminates, returning you to your prompt of the parent shell, in the directory you started from



Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

```
(cd /c && /a/helloworld)
```

Reducing memory usage: To avoid having the subshell waste memory while hello world executes, call `helloworld` via `exec`:

```
(cd /c && exec /a/helloworld)
```

[Thanks to [Josh and Juliano](#) for giving tips on improving this answer!]

edited May 23 '17 at 11:33



Community ♦

1 1

answered Apr 24 '09 at 15:46



[David Schmitt](#)

44.5k 26 107 157

1 Any way to pass arguments to this shell? As in \$1, and \$2? – [finiteloop](#) May 22 '11 at 0:34

2 @segfault: The subshell has complete access to the surrounding scope. – [David Schmitt](#) May 23 '11 at 8:26

Seems like it is temporarily in that directory anyway, isn't it? – [dhein](#) Jan 18 '16 at 9:29

1 You can pass all arguments by doing \$*, @\$ or "\$@" (if you want arguments to respect double quotes) – [Marcel Valdez Orozco](#) Nov 29 '16 at 1:38

Will it work if I add this line to /etc/rc.d/rc.local ? – [Pratik Patil](#) May 15 '18 at 8:12

Similar to [David Schmitt](#)'s answer, plus Josh's suggestion, but doesn't leave a shell process running:

88

```
(cd /c && exec /a/helloworld)
```

This way is more similar to how you usually run commands on the shell. To see the practical difference, you have to run `ps ef` from another shell with each solution.

edited May 23 '17 at 12:03



Community ♦

1 1

answered Apr 24 '09 at 16:05



[Juliano](#)

27k 11 57 71

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook

sh -c 'cd /c && ../a/helloworld'

18

edited May 2 '15 at 23:38



Mark Amery

68.7k 32 266 315

answered Apr 24 '09 at 15:49



mihi

5,251 30 39

1 Used this for FreeBSD's jexec to execute a command inside jail in specified working directory. – [Marián Černý](#) Aug 15 '17 at 5:31

14 wow ur lucky they even let you use a computer in jail – [Matt Wilde](#) Mar 9 '18 at 20:36

@MattWilde freebsd.org/cgi/man.cgi?jail – [mihi](#) Mar 9 '18 at 21:29

An option which doesn't require a subshell and is built in to bash

18

(pushd SOME_PATH && run_stuff; popd)

Demo:

```
$ pwd
/home/abhijit
$ pushd /tmp # directory changed
$ pwd
/tmp
$ popd
$ pwd
/home/abhijit
```

edited Nov 29 '16 at 1:47



Marcel Valdez Orozco

2,690 19 23

answered Jul 13 '15 at 19:28



Loren

2,235 2 20 36

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH

Google

Facebook

- 5 Still, I think `pushd "${SOME_PATH}" && run_stuff; popd` is better than the current answer, since the `pushd/popd` semantics were specifically designed for this situation of going into some directory and then coming back to the original one. – [Marcel Valdez Orozco](#) Nov 29 '16 at 1:40

How does it work as defined as alias and I need to pass a param? – [BeC](#) Apr 7 '17 at 8:37

I think you'd need to write a shell script to which you would pass the parameter that would execute the series of commands as you can't pass a parameter in the middle of an alias. If you need help writing that, you should ask a separate question and reference this one. Once you have the shell script, you could write an alias to call your new script. – [Loren](#) May 8 '17 at 19:31



11



I always think UNIX tools should be written as filters, read input from stdin and write output to stdout. If possible you could change your helloworld binary to write the contents of the text file to stdout rather than a specific file. That way you can use the shell to write your file anywhere.

```
$ cd ~/b
```

```
$ ~/a/helloworld > ~/c/helloworld.txt
```

answered Apr 24 '09 at 17:15



[Luther Blisset](#)

119 2

- 6 +1 for being right, although the answer is only peripherally an answer. – [David Schmitt](#) Apr 25 '09 at 8:04



5



Just change the last "&&" into ";" and it will cd back no matter if the command fails or succeeds:

```
cd SOME_PATH && run_some_command ; cd -
```

edited Mar 29 '16 at 8:10



[Vinoth Krishnan](#)

2,745 6 22 32

answered Mar 29 '16 at 8:07



[johnnybravo](#)

51 1 1

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

back to /D.

Here's a bash shell script example:

```
#!/bin/bash
cd /c
/a/helloworld
```

answered Apr 24 '09 at 15:46



[Jin Kim](#)

7,291 13 44 74

If you always want it to go to /C, use an absolute path when you write the file.

1

answered Apr 24 '09 at 15:40



[Tom Ritter](#)

84.7k 28 125 163

why not keep it simple

1

```
cd SOME_PATH && run_some_command && cd -
```

the last 'cd' command will take you back to the last pwd directory. This should work on all *nix systems.

answered Jan 2 '15 at 3:27



[Sahil](#)

441 5 11

1 It is not going to bring you back if run_some_command fails. – [Anton Daneyko](#) Jan 5 '15 at 15:28

You are write @mezhaka, should have considered that :) – [Sahil](#) Jan 6 '15 at 4:55

Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook

0

```
#include <unistd.h>
int main()
{
    if(chdir("/c") < 0 )
    {
        printf("Failed\n");
        return -1 ;
    }

    // rest of your program...
}
```

answered Apr 24 '09 at 16:16



Harold

220 2 6

He wants to do that in a shell-script, and not in a C. Also, it would be a horrible idea to subprocess the binary file. – user689383 Jan 10 '15 at 3:09

from the current directory provide the full path to the script directory to execute the command

0

```
/root/server/user/home/bin/script.sh
```

answered Jun 20 '17 at 10:23



Ravi Bhushan

35 2

That doesn't change the working directory at all - I think you're answering a different question to the one that was asked. – Toby Speight Jun 20 '17 at 10:53

Came here to find an answer to the question you answered. Thanks! lol – kfrncs Jul 5 '18 at 16:00

Join **Stack Overflow** to learn, share knowledge, and build your career.

[Email Sign Up](#)

OR SIGN IN WITH



Google

