

The 'chmod' Command

By: Nic Stephens and Kelly O'Leary



What is it?

- 💧 A Unix/Linux command
- 💧 Change Mode
- 💧 Changes permissions of a given file

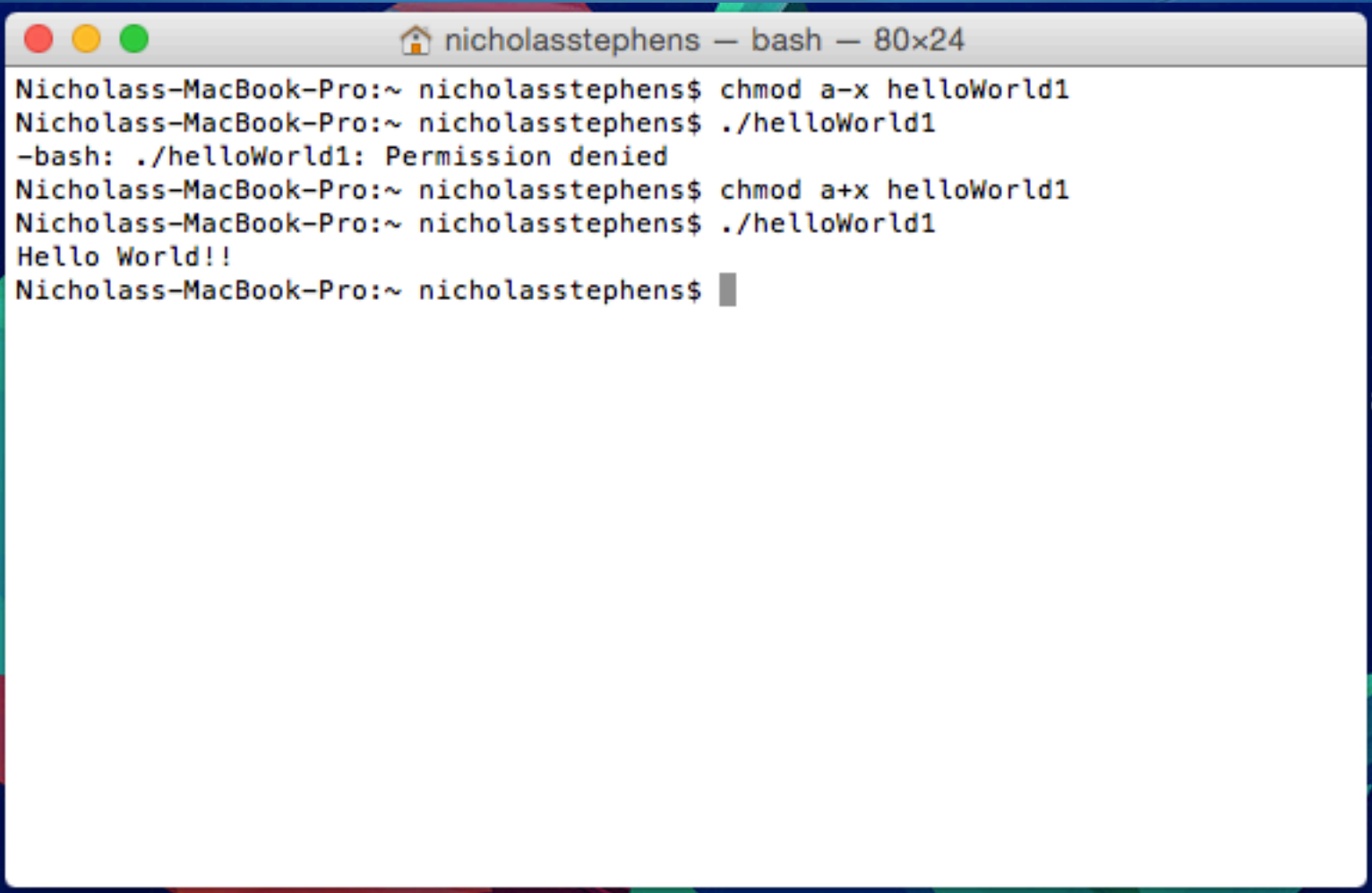
Symbolic Mode

Type of User	Letter/Operator
User who owns it	u
Users in the file's Group	g
Other users not in the file's Group	o
All users	a
Add permission to list	+
Remove permission from list	-
Makes the permission the only permission to the file	=

Symbolic Mode

Permission	Letter
Read	r
Write	w
Execute (Access for Directories)	x
Execute for Directories	X
Set user or group ID on execution	s
Save program text on swap device	t
Gives all permissions that owner has	u
Gives all permissions that users in the file's Group has	g
Gives all permissions that others not in the Group have	o

Example



```
Nicholass-MacBook-Pro:~ nicholasstephens$ chmod a-x helloWorld1
Nicholass-MacBook-Pro:~ nicholasstephens$ ./helloWorld1
-bash: ./helloWorld1: Permission denied
Nicholass-MacBook-Pro:~ nicholasstephens$ chmod a+x helloWorld1
Nicholass-MacBook-Pro:~ nicholasstephens$ ./helloWorld1
Hello World!!
Nicholass-MacBook-Pro:~ nicholasstephens$
```

The image shows a macOS terminal window titled "nicholasstephens — bash — 80x24". The window contains a sequence of commands and their outputs. First, the user runs `chmod a-x helloWorld1`, then `./helloWorld1`, which results in a "Permission denied" error. Next, the user runs `chmod a+x helloWorld1`, and finally `./helloWorld1` again, which successfully outputs "Hello World!!". The prompt `Nicholass-MacBook-Pro:~ nicholasstephens$` is shown at the end of the last line.

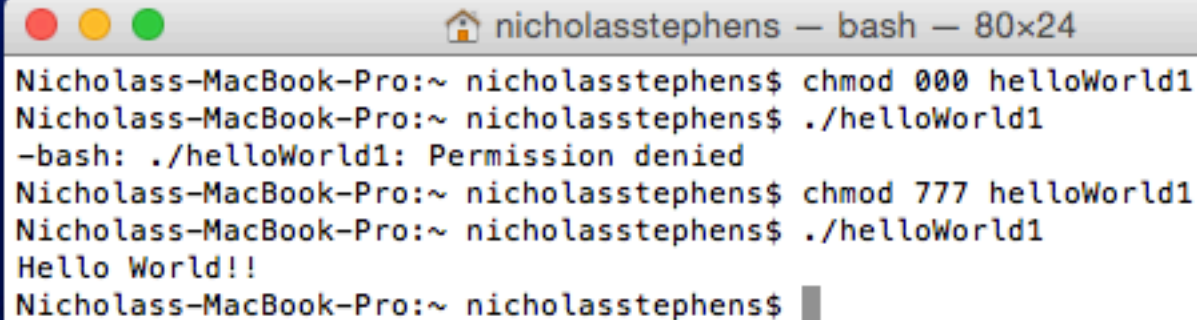
Numeric Mode

- ◆ **First digit (Optional)**- attributes for set user ID (4), group ID (2), and save text image/sticky flags (1)
- ◆ **Second digit**- sets permissions for the owner of the file: Read (4), Write (2), and Execute (1)
- ◆ **Third digit**- sets permissions for users in the file's group: Read (4), Write (2), and Execute (1)
- ◆ **Fourth digit**- sets permissions for users NOT in the file's group: Read (4), Write (2), and Execute (1)

Numeric Mode

- ◆ Add the digits up if there are multiple permissions for a single user
 - ◆ Ex: To put permissions on a file to read, write, and execute, you add up the numbers– read=4, write=2, execute=1.
 $4+2+1=7$, so you type the number 7.
- ◆ Type 0's to remove permissions

Example



A terminal window titled "nicholasstephens — bash — 80x24" with standard macOS window controls (red, yellow, green buttons). The terminal shows a sequence of commands and their outputs:

```
Nicholass-MacBook-Pro:~ nicholasstephens$ chmod 000 helloWorld1
Nicholass-MacBook-Pro:~ nicholasstephens$ ./helloWorld1
-bash: ./helloWorld1: Permission denied
Nicholass-MacBook-Pro:~ nicholasstephens$ chmod 777 helloWorld1
Nicholass-MacBook-Pro:~ nicholasstephens$ ./helloWorld1
Hello World!!
Nicholass-MacBook-Pro:~ nicholasstephens$ █
```


Options

- ◆ -R → causes all files and directories within the file/directory whose permissions are being changed to take those permissions
- ◆ -f, --silent, --quiet → suppress most error messages
- ◆ -v, --verbose → output a diagnostic for every file processed
- ◆ --help → display help and exit

Works Cited

"Chmod." *Man Page*. Web. 3 Sept. 2015. <<http://ss64.com/bash/chmod.html>>.