# NDG Linux Unhatched - NDG Linux Unhatched

content.netdevgroup.com/contents/unhatched/400

## Changing File Permissions

The `chmod` command is used to change the permissions of a file or directory. Only the root user or the user who owns the file is able to change the permissions of a file.

### Consider This

Why is the command named `chmod` instead of `chperm`? Permissions used to be referred to as *modes of access*, so the command `chmod` really means ***ch**ange the **mod**es of access*.

There are two techniques for changing permissions with the `chmod` command: *symbolic* and *octal*. The symbolic method is good for changing one set of permissions at a time. The octal or numeric method requires knowledge of the octal value of each of the permissions and requires all three sets of permissions (user, group, other) to be specified every time. For the sake of simplicity, only the symbolic method will be covered. To learn more about the octal method check out NDG Linux Essentials!

### Follow Along

Use the following command to switch to the `Documents` directory:

```
sysadmin@localhost:~$ cd ~/Documents
```

## The Symbolic Method

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

To use the symbolic method of `chmod` first indicate which *set* of permissions is being changed:

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

| Symbol | Meaning |
|---|---|
| `u` | User: The user who owns the file. |
| `g` | Group: The group who owns the file. |
| `o` | Others: Anyone other than the user owner or member of the group owner. |
| `a` | All: Refers to the user, group and others. |

Next, specify an action symbol:

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

| Symbol | Meaning |
|---|---|
| `+` | Add the permission, if necessary |
| `=` | Specify the exact permission |
| `-` | Remove the permission, if necessary |

After an action symbol, specify one or more permissions to be acted upon.

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

| Symbol | Meaning |
|---|---|
| `r` | read |
| `w` | write |
| `x` | execute |

Finally, a space and the pathnames for the files to assign those permissions.

```
chmod [<SET><ACTION><PERMISSIONS>]... FILE
```

The file `hello.sh` used in the examples on the previous page is a script. A script is a file that can be executed, similar to a command:

```
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rw-r--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

However currently, the execute permission is not set for any of the permission groups:

```
-rw-r--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

Attempting to execute this script using the following syntax fails:

```
sysadmin@localhost:~/Documents$ ./hello.sh
-bash: ./hello.sh: Permission denied
```

Since the system is currently logged in as the `sysadmin` user, and `sysadmin` is the owner of this file, giving the user owner the execute permission should allow you to execute this script. Using the `chmod` command with the `u` character to represent the user owner permission set, the `+` character to indicate a permission is being added, and the `x` character to represent the execute permission, the command should be executed as follows:

```
sysadmin@localhost:~/Documents$ chmod u+x hello.sh
```

No output indicates the command succeeded. Confirm by checking the permissions using the `ls -l` command:

```
sysadmin@localhost:~/Documents$ ls -l hello.sh
-rwxr--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

The user owner now has the execute permission listed:

```
-rwxr--r-- 1 sysadmin sysadmin 647 Dec 20  2017 hello.sh
```

Finally, attempt to execute the script again. Use the command syntax shown below:

```
./hello.sh
```

```
sysadmin@localhost:~/Documents$ ./hello.sh
 ______________
( Hello World! )
 --------------
        \
         \
           <(^)
            ( )
```

### Consider This

Notice that to execute the script in the previous example, a `./` character combination was placed before the script name.

```
./hello.sh
```

This indicates the "command" should be run from the current directory.