

This could be set using `chmod` as follows:

```
$ chmod u=rwx g=rw o=r filename
```

Here:

- ▶ `u` – specifies user permissions
- ▶ `g` – specifies group permissions
- ▶ `o` – specifies others permissions

Use `+` to add permission to a user, group, or others and use `-` to remove the permissions.

Add the executable permission to a file, which is already having the permission `rwx rw- r-` as follows:

```
$ chmod o+x filename
```

This command adds the `x` permission for others.

Add the executable permission to all permission categories, that is, for user, group, and others as follows:

```
$ chmod a+x filename
```

Here `a` means all.

In order to remove a permission, use `-`. For example:

```
$ chmod a-x filename
```

Alternatively, permissions can also be denoted by three-digit octal numbers in which each of the digits corresponds to user, group, and other in that order.

Read, write, and execute permissions have unique octal numbers as follows:

- ▶ `r--` = 4
- ▶ `-w-` = 2
- ▶ `--x` = 1

We can get the required combination of permissions by adding the octal values for the required permission sets. For example:

- ▶ `rw-` = 4 + 2 = 6
- ▶ `r-x` = 4 + 1 = 5

The permission `rw- rw- r--` in the numeric method is as follows:

- ▶ $rw- = 4 + 2 + 1 = 7$
- ▶ $rw- = 4 + 2 = 6$
- ▶ $r-- = 4$

Therefore, `rw- rw- r--` is equal to 764, and the command for setting the permissions using octal values is:

```
$ chmod 764 filename
```

There's more...

Let's go through some additional tasks that can be performed for files and directories.

Changing ownership

In order to change ownership of files, use the `chown` command as follows:

```
$ chown user.group filename
```

For example:

```
$ chown slynux.slynux test.sh
```

Here, `slynux` is the user, as well as the group.

Setting sticky bit

The sticky bit is an interesting type of permission applied to directories. By setting the sticky bit, it restricts only the user owning it to delete the files even though group and others may have sufficient permissions.

In order to set the sticky bit, `+t` is applied on a directory with `chmod` as follows:

```
$ chmod a+t directory_name
```

Applying permissions recursively to files

Sometimes it may be required to recursively change the permissions of all the files and directories inside the current directory. This can be done as follows:

```
$ chmod 777 . -R
```

The `-R` option specifies to apply change to a permission recursively.