Permissions

File permissions

Linux for Programmers and Users, Section 3.30
Taken from Changing file permissions

A file has three types of permissions (read, write and execute) and three sets of users (**u**ser (owner), **g**roup and **o**ther (world)) with specific permissions. Only file's owner or the superuser can change a file's permissions. chmod

chmod

```
  Change file access permissions.
```

SYNOPSIS

```
  chmod [-R] mode FILE...
```

-R, –recursive

```
  change files and directories recursively
```

Two ways of expressing mode:

```
  As an assignment, addition or subtraction of privileges for a specified
set of users. First specify the set of users: u = user; g = group; o =
other; a = all. Next specify the operator: +, -, or =. Finally list which
permissions are being changed or set.
```

```
  {u | g | o | a } {+ | - | =} {rwxs}*
```

```
  chmod -R g+w code  # recursively add group write permissions the code
directroy
  chmod o-rwx *      # remove read, write, execute for other on all files
```

```
  Using a three digit octal number assignment. The three digits correspond
to user, group and other. The value of each digit is as if the rwx
permissions were a three digit binary number. (read = 4, write = 2, execute
= 1)
```

```
  Permission Pattern
```

```
  Octal Number
```

```
  rwxr-x–
```

```
  750
```

```
  rw-r—r—
```

```
  644
```

```
  rw-rw-r—
```

```
  664
```

The first approach, with either the + or - operator, is usually preferred when when operating recursively on a directory tree. This is because some file or directories may have special permissions, thus it is better to add or remove permissions rather explicitly setting them. Also directories require the execute bit set to use the directory. Non-executable files should not have the execute bit set.

Directory Permissions

```
  Read directory permission grants the ability to view a file.
  Write directory permission grants the ability to add, change or remove
files from the directory, assuming the file permissions do not conflict.
  Execute directory permission grants the ability to list (ls) the directory
content of search (find) for files in the directory.
  Desirable permission settings include: 755, 750 or 700.
```

Note

A file is as secure as its directory. The execute permission is not as intuitive as the other two. If this permission is removed, you canâ€™t:

```
  cd to the directory.
  use that directory in a pathname.
  create or remove files in the directory.
```

This means that to be able to create or remove files, the directory must have both write and execute permission. Mere write permission is not enough.

setgid

In addition to the permissions set for files, for a directory you can set the “setgid” bit. This will change the way group is assigned to all files created within this directory: all files created will get the gid of the directory, irrespective of the group the creator of the file belongs to. You can set and clear s with

```
chmod u+s,g+s
```

Permissions indicated with s, e.g. rws, means setgid and execute bits are set. Permissions indicated with S, e.g. rwS, means setgid bit is set, but execute bit is not set.

How a Directory Influences File Permissions

Examining only the user category

File	Directory	Significance
r–r–r–	rwxr-xr-x	A write-protected file; can’t be modified but can be removed.
rw-r–r–	r-xr-xr-x	write-protected directory; file can’t be removed but can be modified.
r–r–r–	r-xr-xr-x	A write-protected file and directory; file can’t be modified or removed.
rw-r–r–	rwxr-xr-x	Normal setting; file can be modified and removed.
rw-r–r–	rw-r-xr-x	File canâ€™t be removed even though directory is writable. (An unusual setting)

Note

```
  All permissions can be changed by the owner or superuser.
  For creating and removing files, the directory must have write and execute
permission (5th example).
  To allow other users to read but not write your files, the directory must
have read and execute permission for that user category.
  The absence of execute permission in any of your directories means that
the find command canâ€™t descend that directory to look for files.
```

An Ownership-Permissions Problem

Assumption: romeo and juliet belong to the users group.

$ who am i romeo $ ls -l foo -r-x-w-r-x 1 juliet users 7017 2004-11-14 13:53 foo $ ls -ld . drwxr-xr-x 21 romeo users 8192 2004-11-28 11:40 .

Note: foo is owned by juliet but directory is owned by romeo.

juliet:

```
  can't edit foo without changing the permissions.
  can change permissions (as owner) and then edit foo.
  can’t delete foo (directory write-protected for group).
```

romeo:

```
  an edit or delete foo.
  can’t change permissions of foo.
  can’t display or copy foo.
```

Extended Permissions

Extended permissions are indicated with an “+” at the end of the permissions string. You can modify them with

```
getfacl
```

and

```
setfacl
```