Understanding the File Mode (Permissions)

Andrew Mallett
LINUX AUTHOR AND TRAINER

@theurbanpenguin www.theurbanpenguin.com

Objectives

List, set, and change standard file permissions

Evaluate permissions needed for file operations and diagnose access issues

Manage file ownership

Linux File Systems and Permissions

In general Linux file systems will support permissions; however, non-native Linux filesystems such as FAT do not.

Linux ACLs

Additional permissions can be added via ACLs. ACLS are covered in a later module. The file mode is limited to a single user and a single group; whereas, the ACLs support many entries.

```
$ ls -l /etc/hosts
-rw-r--r--. 1 root root 220 Jan 10 09:56 /etc/hosts

$ ls -l /etc/shadow
----------. 1 root root 970 Jan 10 10:01 /etc/shadow

<file type> <permissions> <link count> <user group> <size> <modified time>

$ stat /etc/hosts

$ stat -c %a /etc/hosts
644
```

Listing File Permissions

Listing files with the -l option we can see more metadata from the file. This includes the file type, permissions, link count, ownership, file size and the last modified time. The command stat can also be used to view this data.

File Types

regular file	directory	link
pipe	block / character	socket

Demo

Using the ls and stat commands we can list permissions

File Permissions in Linux

Read = 4 in octal and 100 in binary
Read a file or list directory content

Write = 2 in octal and 010 in binary
Create or delete files in directories, write to existing files

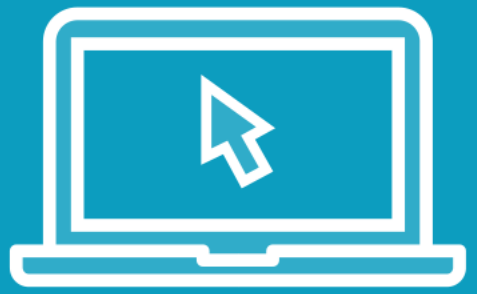
Execute = 1 in octal and 001 in binary
Enter a directory or execute program or script

default permissions for files:
666

default permissions for directories:
777

the current umask value affects default permissions
002

Demo



Working with the umask value and default permissions

Permissions Objects

user: permissions granted to the user owner of the file and no other permissions are applied

group: if the current user does not match the user owner, group membership is checked

others: If the current user id does match the user owner or belong to the group owner then permissions for others are applied

```
$ touch file_perms

$ ls -l file_perms
-rw-rw-r--. 1 vagrant vagrant 0 Jan 15 13:15 file_perms

$ chmod -v 666 file_perms # or

$ chmod -v o+w file_perms
```

Apply Permissions with chmod

The command chmod, change mode, is used to adjust the file permissions. Using the option -v we are able to display both the current and newly assigned permissions. We can use either octal or symbolic notation.

```
$ umask 007

$ mkdir -p upper/{dir1,dir2}

$ touch upper/{dir1,dir2}/file

$ ls -lR upper

$ chmod -vR a+X upper
```

Advanced Symbolic Permissions

Often, it is incorrectly thought that symbolic permissions are simpler and only used when you start your administration career. This is far from the case as we see with -X. The upper-case X is used to set execute only of directories or files where execute is already set in one or more objects.

```
$ umask 007
$ touch another_newfile
$ ls -l another_newfile
$ chmod -v +x another_newfile
$ chmod a+x another_newfile
```

Using All Objects and Omitting the Object

Another misunderstanding the difference between:

chmod +x file and

chmod a+x file omitting the object, chmod applied permissions allowed via the umask. Using -a explicitly, permissions are assigned regardless of the umask

Demo

Setting standard permissions with chmod

Demo

Advanced operations using symbolic permissions

Demo

Ownership of a file can be controlled with the chown and chgrp commands

The minimum permissions needed for a directory is just the execute bit

This allows a user to enter the directory, but not to list the directory contents

Users must know the name of the file they need to access and having read permissions to the files those files

Demo

Minimum permissions is WAY better than maximum permissions:

- Create a new directory
- Assign just execute
- Test and diagnose user access

Summary

Linux file permissions or File Mode

Default permissions:

- file 666
- directory 777

The umask can adjust those defaults

List permissions using

- ls -l
- stat -c %a or A

Using chmod we can set permissions:

- using octals
- or symbols

Ownership set with chown and chgrp

Managing Special Permissions