Andhra Pradesh State Skill Development Corporation

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

AWS CLOUD COMPUTING

LINUX - 2

Basic Linux commands part-2

File permission

We will discuss in detail about file permission and access modes in Unix. File ownership is an important component of Unix that provides a secure method for storing files. Every file in Unix has the following attributes −

- Owner permissions − The owner's permissions determine what actions the owner of the file can perform on the file.
- Group permissions − The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file.
- Other (world) permissions − The permissions for others indicate what action all other users can perform on the file.

The Permission Indicators

While using ls -l command, it displays various information related to file permission as follows −

```
$ ls -l /home/amrood
-rwxr-xr--  1amrood   users 1024  Nov 2 00:10  myfile
drwxr-xr--- 1 amrood   users 1024  Nov 2 00:10  mydir
```

Here, the first column represents different access modes, i.e., the permission associated with a file or a directory.

The permissions are broken into groups of threes, and each position in the group denotes a specific permission, in this order: read (r), write (w), execute (x) −

- The first three characters (2-4) represent the permissions for the file's owner. For example, -rwxr-xr-- represents that the owner has read (r), write (w) and execute (x) permission.
- The second group of three characters (5-7) consists of the permissions for the group to which the file belongs. For example, -rwxr-xr-- represents that the group has read (r) and execute (x) permission, but no write permission.
- The last group of three characters (8-10) represents the permissions for everyone else. For example, -rwxr-xr-- represents that there is read (r) only permission.

Using chmod with Absolute Permissions

The second way to modify permissions with the chmod command is to use a number to specify each set of permissions for the file.

Each permission is assigned a value, as the following table shows, and the total of each set of permissions provides a number for that set.

Number	Octal Permission Representation	Ref
0	No permission	---
1	Execute permission	--x
2	Write permission	-w-
3	Execute and write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read and execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read and write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwx

Here's an example using the testfile. Running ls -1 on the testfile shows that the file's permissions are as follows −

```
$ ls -l testfile
-rwxrwxr--  1amrood   users 1024  Nov 2 00:10  testfile
```

Then each example chmod command from the preceding table is run on the testfile, followed by ls –l, so you can see the permission changes −

```
$ chmod 755 testfile
$ls -l testfile
-rwxr-xr-x  1amrood   users 1024  Nov 2 00:10  testfile

$ chmod 743 testfile
$ls -l testfile
-rwxr---wx  1amrood   users 1024  Nov 2 00:10  testfile

$chmod 043 testfile
$ls -l testfile
----r---wx  1amrood   users 1024  Nov 2 00:10  testfile
```

How to assign a value to string:

A variable is a character string to which we assign a value. The value assigned could be a number, text, filename, device, or any other type of data.

For example, first we set a variable TEST and then we access its value using the echo command −

```
$ TEST="Unix Programming"
$ echo $ TEST
```

It produces the following result.

```
Unix Programming
```

The grep Command:

The grep command searches a file or files for lines that have a certain pattern. The syntax is −

```
$grep pattern file(s)
```

The name "grep" comes from the ed (a Unix line editor) command g/re/p which means “globally search for a regular expression and print all lines containing it”. A regular expression is either some plain text (a word, for example) and/or special characters used for pattern matching.

The simplest use of grep is to look for a pattern consisting of a single word. It can be used in a pipe so that only those lines of the input files containing a given string are sent to the standard output. If you don't give grep a filename to read, it reads its standard input; that's the way all filter programs work −

```
$ ls -l | grep "Aug"
-rw-rw-rw-   1 john  doc     11008 Aug  6 14:10 ch02
-rw-rw-rw-   1 john  doc      8515 Aug  6 15:30 ch07
-rw-rw-r--   1 john  doc      2488 Aug 15 10:51 intro
-rw-rw-r--   1 carol doc      1605 Aug 23 07:35 macros
$
```

The sort Command:

The sort command arranges lines of text alphabetically or numerically. The following example sorts the lines in the food file −

```
$ sort food
Afghani Cuisine
Bangkok Wok
Big Apple Deli
Isle of Java

Mandalay
Sushi and Sashimi
```

Sweet Tooth
Tio Pepe's Peppers
$