Distribute file access with chmod command

Unix and Linux operating systems decentralize access to files and directories using three access parameters, read (read), write (write) and execute (run) to delegate permissions to three groups of objects, including: System owners, administrative groups and users.

***Network* administration - Unix and Linux operating systems decentralize access to files and directories by using three access parameters, read (read), write (write) and execute (run) to delegate to three groups of objects statues, including: system owners, administrative groups and users.**

If you list the properties of a file in detail with the **ls** command with the **-l** switch (for example, **ls -l [filename]**), this command will return information with the form **-rwe-rw-r-** (ie, decentralize read, write and execute to the system owner, grant read and write permissions to the administrative group, and only read permissions to other user objects).

Each of these access privileges corresponds to a value:

1. **read = 4**
2. **write = 2**
3. **execute = 1**

The values ??for some access rights corresponding to each group are added together to form a value between 0 and 7 (can be used to change or decentralize using the chmod command - change mode).

For example, enter the command **chmod 764 [filename]** to grant access to a certain file, in which the value of 764 is generated from:

1. **rwe = 4 (read) + 2 (write) + 1 (execute) = 7**
2. **rw = 4 (read) + 2 (write) = 6**
3. **r = 4 (read) = 4**

You can use **the chmod command** to assign permissions to files and directories, but you should keep in mind the correct chmod command, not the uppercase characters in the command.

The chmod command is often used

Here are some common chmod commands:

1. **chmod 777 filename** : Grant full access to all user objects.
2. **chmod 775 filename** : Grant full access to the system owner and administrative group, the user object can only read (read) and run (execute) the file.

3. **chmod 755 dirname** : Grant full access to the system owner, only allow the administrative group and user objects to read and run the files in the directory.
4. **chmod 700 filename** : Only grant full access to the system owner and block access to all other objects.
5. **chmod 500 dirname** : Do not allow administrators and users to access files in the directory, and limit the read and run system owner permissions to avoid deleting and changing files in this directory.
6. **chmod 660 filename** : Allows system owners and administrators to read, edit, delete and write data to the file, but do not grant access to other users.

See more:

1. Basic Shell commands in Linux
2. 12 best Linux server operating systems