8.10.1 The umask Command

Click one of the buttons to take you to that part of the video.

umask 0:00-0:19

In this lesson, we're going to talk about how to work with default Linux permissions. Now, you may have noticed that whenever you create a new file, or a new directory in the Linux file system, a default set of permissions is automatically assigned to that file or directory for you.

Review Linux Permissions 0:20-2:45

Understand that, by default, Linux assigns read-write to owner; read-write to group; and read-write to others whenever a file is created in the file system. We represent those permissions numerically as 666.

If you create a new directory in the file system, then by default Linux is going to assign read, write, and execute to the directory owner; read, write, and execute to the owning group; and read, write, and execute to all other users in the system.

Be aware that these aren't actually the real permissions that a file or directory will end up with when you create it in the file system. Let's take a look at an example.

Let's suppose I were to create a new directory within my home directory, and we'll name it widgetproject. Within that directory I'm going to create a new file called schedule.odt.

Now, based upon what we just looked at, the widget project directory should have a mode R-W-X, R-W-X, R-W-X, and the schedule.odt file should have a mode of read-write, read-write, read-write--but notice here that this is not the case.

Notice that the widget project directory has a mode of read, write, execute; read, write, execute; and read and execute--which is basically a mode 775. We give read, write, and execute to owner; read, write, and execute to group; and just read and execute to all other authenticated users. For the file schedule.odt we grant read and write permissions to owner and group, and we grant read permissions to others.

Now, you should have noticed that these are not the default permissions that Linux was supposed to assign to this file and directory. Why did this happen? The key thing to remember is that the default permissions Linux wants to assign to files and directories when they're created in the file system are just too liberal.

If you think about it, the default directory mode would allow anybody on the system to enter any directory on the file system and delete any files that they wanted to. Likewise, the default file mode would allow any user on the system to modify anybody else's files that they created. Think about the nightmare situation that would be from a security standpoint.

Use umask to View Permissions 2:45-5:27

To increase the overall security of the system, Linux uses a variable called umask to automatically remove permissions from the default mode that Linux wants to assign whenever a file or directory is created in the file system. The value of umask is a three digit number, and you can view it by running the umask command at the shell prompt.

When you do, you actually see four digits--ignore that first one. For our purposes today, we want to focus on the last three: 002. Now, depending upon your Linux distribution, the default value of umask will be either 022 or 002. I've seen both used.

Remember, each digit represents a numeric permission that will be removed from the default permissions assigned by Linux. So, the first digit in the umask variable represents permissions that will be removed from the file or directory owner, we'll represent as U.

The second digit, as you might guess, represents the permissions that will be removed from the owning group, and the last digit represents permissions that will be removed from others on the system. An example of how umask works is shown here.

Now, the distribution that I ran the umask command on previously had a umask variable of 002, which means the write permission is removed from others. Zero means nothing is removed from the owner, and zero here means that nothing is removed from group.

Let's say we create a new directory in the file system. By default, Linux wants to grant read, write, and execute to every entity--user, group, and owner. But because of the value of umask right here, we subtract the w permission from others.

The resulting mode, the effective permissions that will then be assigned, is shown here. We have read, write, and execute being granted to the owner. Read, write, and execute being granted to group. But because we subtracted the w permission via umask, others get only read and execute.

Likewise, if we create a file in the file system, again Linux wants to assign read and write permissions to that file for owner, group, and others. But because of the value of umask, we're going to remove the write permission from others. The effective permission then is read-write for user and group, and only read for others.

The default value of umask usually works for most of Linux administrators, but there may be situations where you need to either tighten up, most likely, or possibly loosen--not likely--the permissions that are assigned to files or directories when they're created in the file system.

Use umask to Change Permissions 5:28-8:18

To do this, you simply change the value that's assigned to umask. There's two different ways you can do this. First, if you need to make only a temporary change to umask, you simply enter umask at the shell prompt, followed by the numeric permissions that you want subtracted from the default permissions that will be automatically assigned to both files and directories.

For example, if we want to remove the execute permission that's automatically assigned to others whenever a new directory is created, we would specify a 3 in the last spot.

Let's further suppose that we want to remove the write permission that's automatically assigned to group whenever you create a new file or directory in the file system. We want to not touch owner; we want the default permissions assigned to owner to remain intact. So, we'll use a 0 for the first value in the mode.

This will cause the write permission, 2, to be removed from group upon creation of a file or directory. And we'll also remove write, which has value 2; and execute, which has a value of 1 which, summed together, equal three, from others. This will effectively disallow anyone from entering a new directory, except for the directory owner or members of the owning group.

Now, with this new value of umask--remember, it's 023--I've created a new folder. Within that folder I created a new file at the shell prompt. Now, notice now that the effective permissions that have been assigned to that directory and folder are different than they were before. We still have a 0 for user. Therefore, no permissions are subtracted from the default permissions for the directory, or for the file.

Now we're subtracting the write permission from group, so when we created the directory, group did not get write permissions to that directory. And when we created the file, group did not get write permissions to that file. We also subtracted read and write from others. So, when we created the temp files directory, others received read, but they don't receive execute anymore. Likewise, when we created the file, others also just receive read permissions to that file.

This method for modifying the umask variable works great, but be aware that it is not persistent. If I were to restart the system, then umask would revert to its original values, and that's because umask is usually automatically set each time the system boots using the umask parameter in either of these files: either in the /etc/profile file, or the /etc/login.defs file, depending upon which distribution you're using.

So, if you want to make your change to umask permanent, you need to go in and edit the appropriate configuration file in a text editor and set the value of umask to your desired value.

Summary 8:19-8:33

That's it for this lesson. In this lesson we discussed the role and function of umask, and its effect on the default permissions that Linux wants to assign to new files and new directories when they're created in the file system. We also discussed how to modify this behavior by setting the value of umask using the umask command.
