1. **File permissions and ownership**

Permissions are set on files and directories to prevent access by unauthorized users. Users on the system are categorized in to three distinct classes:

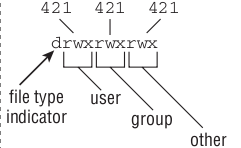
|  |  |
| --- | --- |
| Permission class | Description |
| User (u) | Owner of the file or directory. Usually the creator of the file or directory is the owner of it |
| Group (g) | A set of users that need identical access on files and directories that they share. Group information is maintained in the /etc/group file and users are assigned to groups according to shared file access needs. |
| Others (o) | All other users that have access to the system except the owner and group members. Also called public. |

**Permission Types**

Permissions control what actions can be performed on a file or directory and by whom. There are four types of permissions defined in the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **Permission type** | **Symbol** | **File** | **Directory** |
| Read | R | Displays file contents or copies content to another file | Displays contents with the ***ll*** command |
| Write | W | Modifies file contents | Creates, removes, renames files and sub directories |
| Execute | X | Execute a file | **cd** in a directory |
| Access denied | - | none | none |

The output of the ***ll*** command lists files and directories along with their type and permissions settings. This information is shown in the first column of the output where 10 characters are displayed. The first character indicates the type of file : d for directory, - for regular file, l for symbolic link, c for character device file, b for block device file , n for named pipes, s for socket and so on. The next nine characters – three groups of three characters- show read (r), write (w), execute (x) or none (-) permissions for the three user classes: user, group and others:



**Changing access permissions**

The **chmod** command is used to modify access permissions on files and directories, **chmod** can be used by root or file owner, and can modify permissions specified in one of two methods: symbolic or octal.

**Symbolic notation** uses a combination of letters and symbols to add revoke or assign permissions to each class of users. The following examples will provide an understanding on setting permissions.

We will create a file called john and we will change permissions on it in the below examples:

**# touch john**

**# ls -la**

total 12

drwxr-xr-x 2 root root 4096 Mar 3 16:16 .

drwxr-xr-x 26 root root 4096 Mar 3 16:16 ..

-rw-r--r-- 1 root root 0 Mar 3 16:16 john

The default permissions for user root while creating a file are : user : read + write, group: read, others: read, the owner of the file is root and the group root as seen in columns 3 and 4 of the **ls –la** command output.

Now let’s add write and execute permissions to others and revoke the read from the group in one command using **chmod:**

**# chmod o+wx,g-r john**

**# ls -l**

total 0

-rw----rwx 1 root root 0 Mar 3 16:16 john

After these changes let’s add execute permissions to user root:

**# chmod u+x john**

**# ls -la**

total 12

drwxr-xr-x 2 root root 4096 Mar 3 16:16 .

drwxr-xr-x 26 root root 4096 Mar 3 16:16 ..

-rwx---rwx 1 root root 0 Mar 3 16:16 john

**Homework: read the man pages for the chmod command and practice!**

**The octal notation** uses a three digit numbering system that ranges from 0 to 7 to specify permissions for the three user classes. Octal values are presented below:

0 --- no permission

1 --x execute

2 -w- write

3 -wx write and execute

4 r-- read

5 r-x read and execute

6 rw- read and write

7 rwx read, write and execute

Octal digit values can be added together to make Symbolic Notations:

(4=r)+(1=x)==(5=r-x)  
(4=r)+(2=w)==(6=rw-)  
(4=r)+(2=w)+(1=x) == (7=rwx)

Example :

777 = "-rwxrwxrwx" = rwx for all

754 = "-rwxr-xr--" = rwx for owner, r-x for group, r-- for other

124 = "--x-w-r--" = x for owner, w for group, r for other

Let’s change the permissions to john file to user: read, group write and execute and others set to no permissions.

**# chmod 430 john**

**# ls -la**

total 12

drwxr-xr-x 2 root root 4096 Mar 3 16:16 .

drwxr-xr-x 26 root root 4096 Mar 3 16:16 ..

-r---wx--- 1 root root 0 Mar 3 16:16 john

As you can see the changes have been made to the request.

**Setting default permissions**

The system assigns default permissions to a file or directory when it’s created. Default permissions are calculated based on the **umask** (user mask) permission values subtracted from a pre default value called initial permissions.

**# id**

uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

**# umask**

0022

The –S option at the **umask** command displays the current umask in symbolic notation:

**# umask -S**

u=rwx,g=rx,o=rx

The umask (default permission) is set for files by subtracting out of 666 the value of the umask 022 resulting in 644 (user: read and write, group: read, others: read) and for directories by subtracting out of 777 resulting 755 for the directories created.

User root has the default umask of 022 on the system, and other users have the default umask set to 002. The default value can be changed.

**Homework: How can we change the default value of the umask for a user on a server?**

**Changing file ownership and group membership.**

In Linux, every file and directory has an owner associated with it. By default, the creator becomes the owner. The ownership can be changed if required, and allocated to some other user. Similarly, every user is a member of one or more groups. A group is a collection of users with exact same privileges. By default the owner’s group is assigned to a file or directory.

**# ls -la**

total 12

drwxr-xr-x 2 root root 4096 Mar 3 17:01 .

drwxr-xr-x 26 root root 4096 Mar 3 16:16 ..

-rw-r--r-- 1 root root 0 Mar 3 17:01 bla

-r---wx--- 1 root root 0 Mar 3 16:16 john

**# chown john john**

**# chgrp john john**

**# ls -al**

total 12

drwxr-xr-x 2 root root 4096 Mar 3 17:01 .

drwxr-xr-x 26 root root 4096 Mar 3 16:16 ..

-rw-r--r-- 1 root root 0 Mar 3 17:01 bla

-r---wx--- 1 john john 0 Mar 3 16:16 john

Using the **chgrp** and **chown** commands I changed the owner and the group of the file john to user and group john. Read the manual pages for the two commands and get yourself familiarized to them.