Lab-3 09-19-2020

Start a script session to record your activities

Script glitched and I had to delete it, it looped when I did cat into some ridiculous 10 digit number of bytes.

Your umask is set when you first log into the system. By default, the system sets it in the **/etc/profile** file.The system checks your User/Group IDs to see if it's greater than 199. If so, it sets your umask to 002 otherwise it sets your umask to 022

1.) View your umask

$ umask

> Write the values and describe what you see.

0002, which are four digits that represent file permissions

3. Go into your **~/homework/week-3/lec-3** directory.

4. Verify that you are in your lec-3 directory before you execute the following commands. Record the output of ls -ld dirx.

$ mkdir dirx

$ ls -ld dirx

drwxrwxr-x. 2 cs45aa05 cs45aa05 6 Sep 24 05:10 dirx

> drwxr-xr-x

5. Create an empty file

$ touch xfile

$ ls -l xfile

> who owns xfile?

cs45aa05

> What is the size of xfile?

0

> What are the permissions of xfile?

-rw-rw-r--.

> What are the owners' permission for xfile?

Read and write

> What are the permissions for the group owner of xfile?

Read and write

> If you are not the owner and not in the group, what permissions do you get?

Read

6. Change your umask to 033

$ umask 033

7. Create another file called zfile

$ touch zfile

$ ls -l zfile

> What are the permissions of zfile?

-rw-r--r--.

> Why does zfile have those permissions?

With umask 033, the group and the other users only get read permissions

8. Create a directory called 'zdir'

$ mkdir zdir

$ ls -ld zdir

> What are the permissions of zdir

drwxr--r--.

> Why does zdir have those permissions

Those are the current default directory permissions

What does the 'x' permission allow on a directory?

Allows for the access to files in the directory

What does the 'w' permission allow on a directory?

Allows for the ability to write new files in a directory

What does the 'r' permission allow on a directory?

Allows for you to list the contents of a directory

9. What groups does mr-tester belong to?

$ groups mr-tester

mr-tester : mr-tester

Search for mr tester in the /etc/group file.

$ grep mr-tester /etc/group

mr-tester:x:1047:

$ id -a mr-tester

> Record the group ID for mr-tester

uid=1045(mr-tester) gid=1047(mr-tester) groups=1047(mr-tester)

10. How can I change my current group id during this login session? What is the difference using the '-' and without the '-' . You will change your current real groupID to the new group or if no new group name is listed, you get the default in /etc/passwd file.

$ newgrp - cs45

$ ps

$ exit

$ ps

PID TTY TIME CMD

24968 pts/2 00:00:00 bash
27274 pts/2 00:00:00 ps

$ newgrp cs45

$ ps

$ exit

$ ps

Asks for password

The /etc/group file has group account information.

The /etc/gshadow file has the group password information

The /etc/passwd file has user account information

The /etc/shadow file has user encrypted password information

11. Use the chgrp command to assign your file to another group that you belong to.

$ touch gxfile

$ ls -l gxfile

> What is the group this file belong to?

cs45aa05

$ chgrp wheel gxfile

> did this work?

No operation was not permitted

$ exit

12. Try to give your group ownership to cs45.

$ touch mr-tfile

$ ls -l mr-tfile

> Record your output

-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 22:26 mr-tfile

$ chgrp cs45 mr-tfile

> Did this work?

nope

> Why not?

The operation was not permitted

13. Test to see if mr-tfile is a regular file. If the test is successful, you will get a return or exit status of zero '0'. The test command is used to test file types and compare values.

$ test -f mr-tfile

$ echo $?

> What is your output?

$

$ mkdir mr-dir

$ ls -ld mr-dir

$ test -d mr-dir

$ echo $0

14. Let's have some fun with the test command.

{ lt = less than ; gt = greater than; ne = not equal; ge = greater or equal , && = logical AND . If the left side is true ...do the right side}. What is the purpose of the '**$?'**

$ test 4 -lt 5

$ echo $?

0

$ test 4 -lt 5 && echo "I got this!"

-bash: !": event not found

$ test 2 -ne 3 && echo "Rock on"

Rock on

$ test -d mr-dir && echo "mr-dir is a directory"

mr-dir is a directory

Changing Permissions

> chmod [option] ... MODE,.... FILE

u = user/owner

g = group

o = others

a = all (user/owner, group and others)

Permissions : read(r) octal value = 4

write(w) octal value = 2

execute(x) octal value = 1

set UID(SUID) 'u+s' octal value = 4

set GID(SGID) 'g+s' octal value = 2

set Sticky Bit 'o+t' octal value = 1

Usage

15. Create a directory called permtest,

$ mkdir permtest

$ cd permtest

$ touch feedback1 feedback2 feedback3

$ ls -lh

> Record the permissions

-rw-rw-r--

-rw-rw-r--
-rw-rw-r--

16. vi feedback1 and add two lines and safe the file.

echo "I am : $0 "

echo "Practice harder!:)"

17. Give the file execute permissions

$ chmod +x feedback1

$ ls -lh

18. Use the symbolic options(u,g,o,a) to give feedback2 read,write for owner, read for group and nothing for others.

-rwxrwxr-x. 1 cs45aa05 cs45aa05 44 Sep 29 22:41 feedback1
-rw-r-----. 1 cs45aa05 cs45aa05 0 Sep 29 22:34 feedback2
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 22:34 feedback3

19. Use the symbolic options to subtract write from group and others on feedback2

-rw-r-----. 1 cs45aa05 cs45aa05 0 Sep 29 22:34 feedback2

20 Use the Octal permissions to give read,write,execute to owner of feedback3, read and execute to group of feedback3 and read only for others.

-rwxr-xr--. 1 cs45aa05 cs45aa05 0 Sep 29 22:34 feedback3

21. Make a directory called 'dir-tester'

$ mkdir dir-tester

22. Make another directory in dir-tester called dir-test2

23. Create 4 files in dir-test2 called 'tester-file{1..4}'

$ touch dir-tester/dir-test2/tester-file{1..4}

$ ls dir-tester/dir-test2/*

-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file1

-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file2
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file3
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file4

24. Recursively do a long listing of dir-tester and all contents below

$ ls -lR dir-tester

> Record the permissions

.:

total 0
drwxrwxr-x. 2 cs45aa05 cs45aa05 86 Sep 29 23:02 dir-test2

./dir-test2:
total 0
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file1
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file2
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file3
-rw-rw-r--. 1 cs45aa05 cs45aa05 0 Sep 29 23:02 tester-file4

25. Recursively change the permissions of all the content of dir-tester --all files and directory below to 600

$ chmod -R 600 dir-tester

> Verify the output by doing the ls -lR

> Record the permissions you see

.:
total 0
drw-------. 2 cs45aa05 cs45aa05 86 Sep 29 23:02 dir-test2

./dir-test2:
ls: cannot access ./dir-test2/tester-file1: Permission denied
ls: cannot access ./dir-test2/tester-file2: Permission denied
ls: cannot access ./dir-test2/tester-file3: Permission denied
ls: cannot access ./dir-test2/tester-file4: Permission denied
total 0
-????????? ? ? ? ? ? tester-file1
-????????? ? ? ? ? ? tester-file2
-????????? ? ? ? ? ? tester-file3
-????????? ? ? ? ? ? tester-file4

Changing ownership - Must be super user

chown [new owner] file_name or Directory_Name

$ chown hcampbell file_name

Recursively change owner

$ chown -R <New Owner> Directory_name