Lab1: Access Control
---Unix

Unix File Hierarchy

The Unix file system is organized as a hierarchy with the root (/) directory at the highest level. Each directory may contain subdirectories and files. Typically, some of the directories that may occur under the root are usr, bin, sbin, home, var, boot, dev, etc. In Figure 3, user1 and user2 are sub-directories under home. hello.txt is a plain-text file and link_hello is a linking file that points to hello.txt. In order to access the file /test/temp/hello.txt, the system begins its search from the root(/) folder and then to test and temp folders consecutively and then finally the file hello.txt.

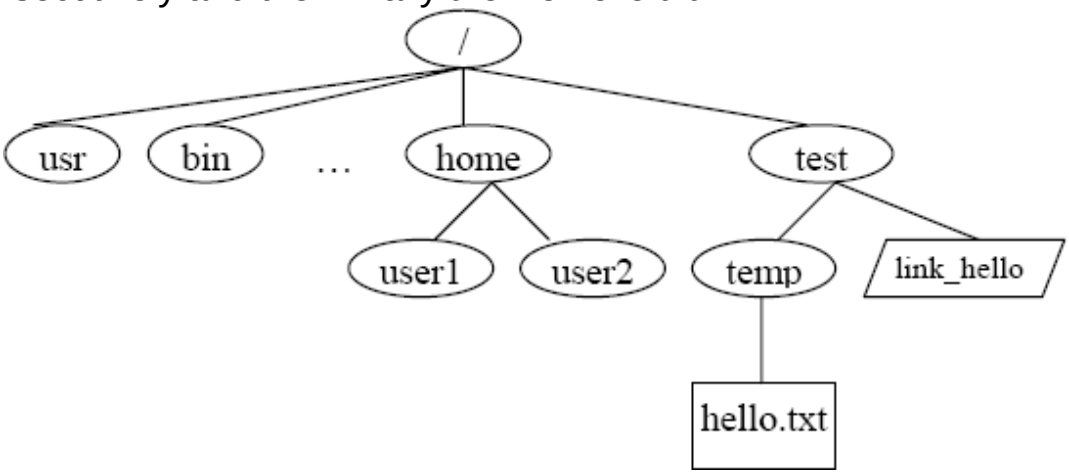

Figure 3

Ownership and Permissions

Ownership of files in UNIX can be viewed in one of three ways: owner (creator), group or others. Using this simple notion of ownership access to files can be controlled by associating unique user ID (UID) and group ID (GID) with twelve permission bits for each file as shown below.

Permission Bits

Extra			Owner			Group			Others		
su	sg	t	r	w	x	r	w	x	r	w	x

Typically these bits are divided into three sets of three bits and three extra bits as shown in table below. r, w and x bits stand for read, write and execute bits for each of the owner, group and others permissions. su, sg and t stand for set_user_id, set_group_id and sticky bits. These 4 sets of bits are often represented in their octal digits. For example, "100 111 101 101" is represented as "4755." When the su bit is set, whosoever executes the file, the UID of the process will be the owner of the file.

Unix Lab Procedures

1 Setting up File Structure and User Space. The objective of this exercise is to setup the file hierarchy structure and the users that are required for the exercises in this section. The su command is used to switch users.

a. Login as root
b. Use useradd command to create two new users user1 and user2 as follows:
    i. `useradd –m user1 -g users -p user1`
    ii. `useradd –m user2 -g users -p user2`
c. Check user information with the id command. Note the uid, gid for each output.
    i. `id user1`
    ii. `id user2`
    iii. `id`
d. Create a directory structure
    i. `mkdir /test`
    ii. `mkdir /test/temp`
e. Switch user roles as user1 and then back to root using the su command
    i. `whoami`
    ii. `su user1`
    iii. `su` OR `su root`

f. Create a new file as root user and change group ownership as well as user ownership of the file.
    i. `touch /home/user2/HelloWorld`
    ii. `ls -l /home/user2/HelloWorld` (observe owner and group)
    iii. `chgrp users /home/user2/helloWorld`
    iv. `chown user2:users /home/users/HelloWorld`
    v. `ls -l /home/user2/HelloWorld` (observe owner and group)

2 Questions.
a. Explain what `chgrp` and `chown` do?
b. What do `-g` and `-p` options mean?

3 Differences in File and Folder Permissions. The objective of the following exercises would be to see the differences in file and folder permissions. The chmod command will be used to change file and directory permission to demonstrate the slight differences in permissions for files and directories.
a. Observe the result of ls and cd commands
    a. `cd /`
    b. `ls -l`
    c. `ls -al /home`
    d. Switch to user1 using `su user1`
    e. `ls -al /home/user2` (Can you list directory?)
    f. `cd /home/user2` (Can you change directory?)

b. Change directory permissions of user2 directory and try again as user1.
    a. su root
    b. chmod 740 /home/user2
    c. Repeat above step d to f(Can you list or change directory?)
    d. su root
    e. chmod 750 /home/user2
    f. Repeat above step d to f (Can you list or change directory?)
    g. touch /home/user2/hello12.txt(Can you create new file?)
    h. su root
    i. chmod 770 /home/user2
    j. su user1
    k. Repeat step g. (Can you create new file?)

l. ls -l /home/user2

4 Question. What are the directory permissions for user1, user2 and test directories?

5 Alternative Syntax for chmod Command. You are expected to learn both the ways to use chmod. The access permissions for the file hello.txt is to set the su bit only, allow all access permissions to owner, read and execute rights to the group and only read rights to others. In other works the 12 bit permission required on the file hello.txt is as follows: "100 111 101 100." This can be achieved in several ways using chmod command:

a. chmod 4754 hello.txt
b. chmod u+srwx, g+rx, o+r hello1.txt
c. chmod u=srwx, g=rx, o=r hello2.txt
d. ls –l /home/user2