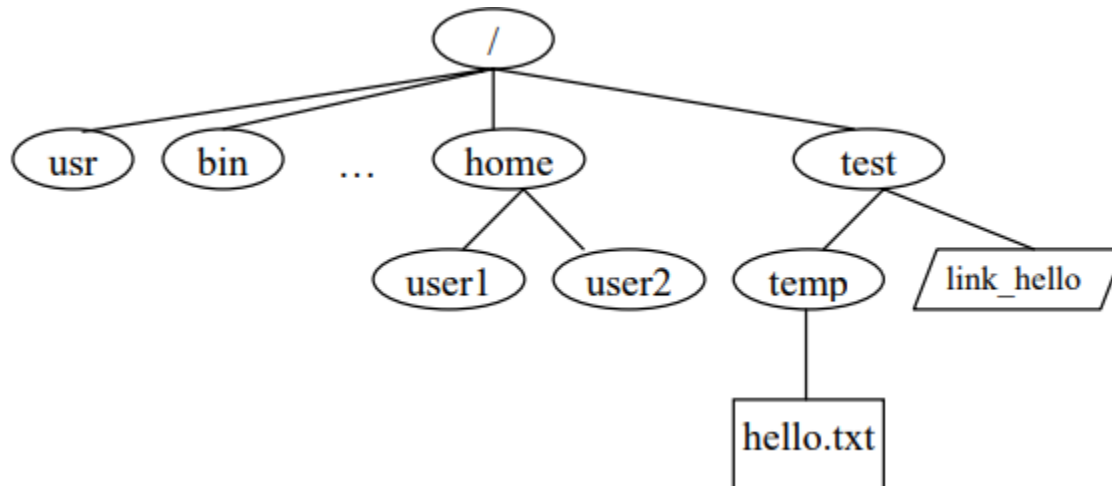


File Hierarchy

The Unix file system is organized as a hierarchy with the root (/) at the highest level from which directories and files may exist. Typically, some of the directories that may occur under the root are *usr*, *bin*, *sbin*, *home*, *var*, *boot*, *dev*, etc, etc. In the figure shown below, *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*.



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 a unique user ID (UID) and group ID (GID) with twelve permission bits for each file. Typically these bits are divided into three sets of three bits and three extra bits as shown in table below. P

Permission Bits											
Extra			owner			group			others		
Su	sg	t	r	w	x	r	w	x	r	W	x

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. Similarly, if sg is set, the GID of the process will be the owner of the file.

Task 1: Setting up user space

1. Login as root

```
studio@mashrur-safir:~$ sudo passwd root
[sudo] password for studio:
New password:
Retype new password:
passwd: password updated successfully
studio@mashrur-safir:~$ su -
Password:
root@mashrur-safir:~#
```

2. Create 2 users, student1 and student2 to the group users

```
root@mashrur-safir:~# useradd -m -G users student1
root@mashrur-safir:~# useradd -m -G users student2
root@mashrur-safir:~# cd ..
root@mashrur-safir:~# cd home
root@mashrur-safir:/home# ls
student1 student2 studio
```

3. Create a file Hello in the student2's home directory

```
root@mashrur-safir:/home# cd student2
root@mashrur-safir:/home/student2# touch hello
```

4. Use the command "ls" to answer who is the owner and what is the group of the file Hello?

```
root@mashrur-safir:/home/student2# ls -l
total 0
-rw-r--r-- 1 root root 0 এপ্রিল  24 07:08 hello
```

File_permissions = -rw-r--r-- 1

U = rw

G = r

O = r

Owner = root

Group = root

5. Change the group of the file Hello to users and change the owner of the file Hello to student2.

```
root@mashrur-safir:/home/student2# chgrp users hello
root@mashrur-safir:/home/student2# chown student2 hello
```

6. Use the command “ls” to answer who is the owner and what is the group of the file Hello?

```
root@mashrur-safir:/home/student2# ls -l
total 0
-rw-r--r-- 1 student2 users 0 এপ্রিল 24 07:08 hello
```

Task 2: File and folder permissions

The objective of the following exercises is to see the differences between file and folder (directory) permissions. The chmod command will be used to change file and directory permission to demonstrate the slight differences in permissions for files and directories.

Change the group of both student1 and student2 using chgrp to users.

```
root@mashrur-safir:/home# chgrp users student1
root@mashrur-safir:/home# chgrp users student2
```

1. Login as root

```
studio@mashrur-safir:~$ sudo passwd root
[sudo] password for studio:
New password:
Retype new password:
passwd: password updated successfully
studio@mashrur-safir:~$ su -
Password:
root@mashrur-safir:~#
```

2. What are the directory permissions for student1's, and student2's home directories?

```
root@mashrur-safir:/home# ls -l
total 12
drwxr-xr-x 2 student1 users 4096 এপ্রিল 24 07:04 student1
drwxr-xr-x 2 student2 users 4096 এপ্রিল 24 07:08 student2
drwxr-xr-x 27 studio studio 4096 এপ্রিল 20 00:05 studio
```

3. Login as student1

```
root@mashrur-safir:/home# su - student1
$ pwd
/home/student1
$
```

4. Can you list the student2's home directory?

```
$ ls /home/student2
hello
```

5. Can you change the working directory to the student2's home directory?

```
$ cd /home/student2
```

```
$ pwd
```

```
/home/student2
```

6. Switch user to root

```
$ su -
```

```
Password:
```

```
root@mashrur-safir:~#
```

7. Change the permission of student2's home directory to 740

```
root@mashrur-safir:~# chmod 740 /home/student2
```

```
root@mashrur-safir:~# ls
```

```
snap
```

```
root@mashrur-safir:~# cd ..
```

```
root@mashrur-safir:/# ls
```

```
bin dev lib libx32 mnt root snap sys var
```

```
boot etc lib32 lost+found opt run srv tmp
```

```
cdrom home lib64 media proc sbin swapfile usr
```

```
root@mashrur-safir:/# cd home
```

```
root@mashrur-safir:/home# ls
```

```
student1 student2 studio
```

```
root@mashrur-safir:/home# ls -l
```

```
total 12
```

```
drwxr-xr-x 2 student1 users 4096 এপ্রিল 24 07:04 student1
```

```
drwxr----- 2 student2 users 4096 এপ্রিল 24 07:08 student2
```

```
drwxr-xr-x 27 studio studio 4096 এপ্রিল 20 00:05 studio
```

8. Switch user to student1. Can you list or change to the student2's home directory?

```
root@mashrur-safir:/home# su - student1
```

```
$ pwd
```

```
/home/student1
```

```
$ ls /home/student2
```

```
hello
```

```
$
```

```
$ cd /home/student2
```

```
-sh: 3: cd: can't cd to /home/student2
```

9. Switch user to root

```
$ su -
```

```
Password:
```

```
root@mashrur-safir:~#
```

10. Change the permission of student2's home directory to 750

```
root@mashrur-safir:~# chmod 750 /home/student2
root@mashrur-safir:~# cd ..
root@mashrur-safir:/# cd home
root@mashrur-safir:/home# ls -l
total 12
drwxr-xr-x 2 student1 users 4096 এপ্রিল 24 07:04 student1
drwxr-x--- 2 student2 users 4096 এপ্রিল 24 07:08 student2
drwxr-xr-x 27 studio studio 4096 এপ্রিল 20 00:05 studio
```

11. Switch user to student1. Can you list or change to the student2's home directory?

```
root@mashrur-safir:/home# su - student1
$ pwd
/home/student1
$ ls -l /home/student2
total 0
-rw-r--r-- 1 student2 users 0 এপ্রিল 24 07:08 hello
$ ls /home/student2
hello
$ cd /home/student2
$ pwd
/home/student2
```

12. Can you create a new file, Hello2, in the student2's home directory?

```
$ touch hello2
touch: cannot touch 'hello2': Permission denied
```

13. Switch user to root

```
$ su -
Password:
root@mashrur-safir:~#
```

14. Change the permission of student2's home directory to 770.

```
root@mashrur-safir:~# chmod 770 /home/student2
root@mashrur-safir:~# cd ..
root@mashrur-safir:/# cd home
root@mashrur-safir:/home# ls -l
total 12
drwxr-xr-x 2 student1 users 4096 এপ্রিল 24 07:04 student1
drwxrwx--- 2 student2 users 4096 এপ্রিল 24 07:08 student2
drwxr-xr-x 27 studio studio 4096 এপ্রিল 20 00:05 studio
```

15. Switch user to student1. Can you create a new file, Hello2, in the student2's home directory?

```
root@mashrur-safir:/home# su - student1
$ pwd
/home/student1
$ touch /home/student2/hello2
$ ls /home/student2
hello hello2
```

16. What does it mean if the folder permission is 764?

Task 3: Permission of a symbolic link file

Unix supports two kinds of link files--a hard link and a symbolic link. A hard link is a file with the actual address space of some ordinary file's data blocks. A symbolic link is just a reference to another file. It contains the pathname to some other file. (It's a sort of shortcut to access a file)

1. Login as root

```
$ su -
Password:
root@mashrur-safir:~#
```

2. Create "home/test/temp" directory

```
root@mashrur-safir:~# ls
snap
root@mashrur-safir:~# cd ..
root@mashrur-safir:/# ls
bin  dev  lib  libx32  mnt  root  snap  sys  var
boot  etc  lib32  lost+found  opt  run  srv  tmp
cdrom  home  lib64  media  proc  sbin  swapfile  usr
root@mashrur-safir:/# mkdir -p home/test/temp
root@mashrur-safir:/# cd home
root@mashrur-safir:/home# ls
student1  student2  studio  test
```

3. In the "home/test/temp" directory, create a new file named source

```
root@mashrur-safir:/home# cd ..
root@mashrur-safir:/# touch /home/test/temp/source
```

4. Create a link file named link_source in the "/home/test" directory pointing to the source in the "/test/temp" directory by using the command "ln -s /home/test/temp/source /home/test/link_source"

```
root@mashrur-safir:/# ln -s /home/test/temp/source /home/test/link_source
```

5. What are the file permissions of the link_source and source?

```
root@mashrur-safir:/home/test# ls -l
total 4
lrwxrwxrwx 1 root root 22 এপ্রিল 24 08:37 link_source -> /home/test/temp/source
drwxr-xr-x 2 root root 4096 এপ্রিল 24 08:25 temp
root@mashrur-safir:/home/test# cd temp
root@mashrur-safir:/home/test/temp# ls
source
root@mashrur-safir:/home/test/temp# ls -l
total 0
-rw-r--r-- 1 root root 0 এপ্রিল 24 08:25 source
```

6. What is the information inside the file link_source?

```
root@mashrur-safir:/home/test/temp# echo "Hello World!" >> source
root@mashrur-safir:/home/test/temp# cat source
Hello World!
root@mashrur-safir:/home/test/temp# cd ..
root@mashrur-safir:/home/test# ls
link_source temp
root@mashrur-safir:/home/test# cat link_source
Hello World!
```

Task 4: Default file permission

Whenever a new file is created a default set of permissions can be assigned to it. Whatever the permissions are, the UNIX system allows the user to filter out unwanted permissions set by default. This default setting can be set by the user using the umask command. The command takes the permissions set during creation of file and performs a bitwise AND to the bitwise negation of mask value. Some common umask values are 077 (only user has permissions), 022 (only owner can write), 002 (only owner and group members can write), etc.

1. Login as root

```
$ su -
Password:
root@mashrur-safir:~#
```

2. Use the command “umask -S” or the man page to learn how to use the command umask.

3. What is the current mask? How is it interpreted?

4. Create a file named testmask1 in the “/test” directory. What are the permissions of the file testmask1? (hint: ls)

```
root@mashrur-safir:~# cd ..
root@mashrur-safir:/# cd home
```

```

root@mashrur-safir:/home# ls
student1 student2 studio test
root@mashrur-safir:/home# cd test
root@mashrur-safir:/home/test# ls
link_source temp
root@mashrur-safir:/home/test# touch testmark1
root@mashrur-safir:/home/test# ls -l
total 4
lrwxrwxrwx 1 root root 22 এপ্রিল 24 08:37 link_source -> /home/test/temp/source
drwxr-xr-x 2 root root 4096 এপ্রিল 24 08:25 temp
-rw-r--r-- 1 root root 0 এপ্রিল 24 09:11 testmark1

```

5. Change the mask to 0077 (current directory), then create a new file named testmask2.

```

root@mashrur-safir:/home/test# umask 0077
root@mashrur-safir:/home/test# touch testmask2

```

6. What are the permissions of the file testmask2?

```

root@mashrur-safir:/home/test# ls -l
total 4
lrwxrwxrwx 1 root root 22 এপ্রিল 24 08:37 link_source -> /home/test/temp/source
drwxr-xr-x 2 root root 4096 এপ্রিল 24 08:25 temp
-rw-r--r-- 1 root root 0 এপ্রিল 24 09:11 testmark1
-rw----- 1 root root 0 এপ্রিল 24 09:12 testmask2

```

7. What is the effect of setting mask value to 0000?

No permissions are masked out.

Files: 666

Folders: 777

So when they are created they will get these maximum permissions.

8. Change the mask back to what you noted in #3

```

root@mashrur-safir:/home/test# umask 0022

```

Task 5: Setuid Bit

As explained in the ownership and permission section, the highest three bits of the permission value of a file represent the setuid bit, setgid bit and the sticky bit. If the setuid bit is set then the uid will always be set to the owner of the file during execution. If the setuid bit is not set then the uid will be the user who executes the process. Similarly, if the setgid bit is set then the gid will be set to the group that owns the file during execution. If the setgid bit is not set then the gid will be the group that executes the process. The sticky bit is set to keep processes in the main memory.

1. Login as root


```
$ su -  
Password:  
root@mashrur-safir:~#
```

2. Use the command “which touch” to find the path to touch command

```
root@mashrur-safir:/home/test# which touch  
/usr/bin/touch
```

3. What are the permissions of touch?

```
root@mashrur-safir:/home/test# ls -l /usr/bin/touch  
-rwxr-xr-x 1 root root 100728 সেপ্টেম্বর 5 2019 /usr/bin/touch
```

4. Change touch’s permissions to 4755.

```
root@mashrur-safir:/home/test# chmod 4755 /usr/bin/touch  
root@mashrur-safir:/home/test# ls -l /usr/bin/touch  
-rwsr-xr-x 1 root root 100728 সেপ্টেম্বর 5 2019 /usr/bin/touch
```

5. Change the permissions of file Hello in the student2’s home directory to 700.

```
root@mashrur-safir:/# chmod 700 /home/student2/hello
```

6. What are the permissions and timestamp of the file Hello? (hint: ls)

```
root@mashrur-safir:/# ls -l /home/student2/hello  
-rwx----- 1 student2 users 0 এপ্রিল 24 07:08 /home/student2/hello
```

7. Switch to student1.

```
root@mashrur-safir:/# su - student1  
$ pwd  
/home/student1
```

8. Use the command “touch” to the file Hello. What is the file Hello’s timestamp?

```
$ touch /home/student2/hello  
$ ls -l /home/student2/hello  
-rwx----- 1 student2 users 0 এপ্রিল 24 10:55 /home/student2/hello
```

9. Switch to root.

```
$ su -  
Password:  
root@mashrur-safir:~#
```

10. Change the touch’s permissions to 0755.

```
root@mashrur-safir:~# chmod 0755 /usr/bin/touch  
root@mashrur-safir:~# ls -l /usr/bin/touch  
-rwxr-xr-x 1 root root 100728 সেপ্টেম্বর 5 2019 /usr/bin/touch
```

11. Switch to student1.

```
root@mashrur-safir:~# su - student1
$ pwd
/home/student1
```

12. Use the command “touch” to the file Hello. Why is permission denied? Why is it possible to do in the #7?

```
$ touch /home/student2/hello
touch: cannot touch '/home/student2/hello': Permission denied
```