Practical 4

Working with Users, Groups and Permissions

Working with Users

#useradd abc: to add new user.

#adduser pqr: to add new user.

#useradd xyz: to add new user.

#passwd abc: to set or change the password of user abc.

#passwd pqr: to set or change the password of user pqr.

#passwd xyz: to set or change the password of user xyz.

```
[root@localhost ~]# useradd abc
[root@localhost ~]# adduser pqr
[root@localhost ~]# useradd xyz
[root@localhost ~]# passwd abc
Changing password for user abc.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# passwd pqr
Changing password for user pqr.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# passwd xyz
Changing password for user xyz.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

#passwd -l pqr: to lock the user account pqr.

#passwd -u pqr: to unlock the user account pqr.

```
[root@localhost ~]# passwd -l pqr
Locking password for user pqr.
passwd: Success
[root@localhost ~]# passwd -u pqr
Unlocking password for user pqr.
passwd: Success
[root@localhost ~]#
```

```
[student@localhost ~]$ su pqr
Password:
su: Authentication failure
[student@localhost ~]$ su pqr
Password:
[pqr@localhost student]$
```

#su abc: to switch user to abc.

#su pqr: to switch user to pqr.

#su xyz: to switch user to xyz.

```
[root@localhost ~]# su abc
[abc@localhost root]$ su pqr
Password:
[pqr@localhost root]$ su xyz
Password:
[xyz@localhost root]$
```

#chfn abc: to add finger information for user abc.

#getent passwd abc: to view finger information of user abc.

```
[root@localhost ~]# chfn abc
Changing finger information for abc.
Name []: student
Office []: froncs
Office Phone []: 8456279415
Home Phone []: 7854612937

Finger information changed.
[root@localhost ~]# getent passwd abc
abc:x:1001:1001:student,froncs,8456279415,7854612937:/home/abc:/bin/bash
```

#userdel xyz: to delete user (home directory of user remains in the system).

#userdel -rf xyz: to delete user along with his home directory.

```
[root@localhost ~]# userdel -rf xyz
[root@localhost ~]# ls /home
abc  pqr  student
[root@localhost ~]#
```

#chage -l student: to display the password aging information for a user student.

```
[root@localhost ~]# chage -l student
Last password change                                    : never
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
```

#cat /etc/group: to display the content of the /etc/group file.

```
[root@localhost ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mem:x:8:
kmem:x:9:
wheel:x:10:

dovenull:x:374:
sshd:x:74:
oprofile:x:16:
tcpdump:x:72:
student:x:1000:student
abc:x:1001:
pqr:x:1002:
```

Working with Groups

#useradd user1: to add a new user.

#useradd user2: to add a new user.

#**useradd user3:** to add a new user.

#groupadd usersgrp: to add a group named usersgrp.

#usermod -g usersgrp user1: assigns usersgrp as primary group to user named user1.

#usermod -g usersgrp user2: assigns usersgrp as primary group to user named user2.

#usermod -G usersgrp user3: assigns usersgrp as secondary group to user named user3.

#id user1: to get detailed information about user1.

#id user2: to get detailed information about user2.

#id user3: to get detailed information about user3.

#cat /etc/group: to display the content of the /etc/group file.

```
[root@localhost ~]# useradd user1
[root@localhost ~]# useradd user2
[root@localhost ~]# useradd user3
[root@localhost ~]# groupadd usersgrp
[root@localhost ~]# usermod -g usersgrp user1
[root@localhost ~]# usermod -g usersgrp user2
[root@localhost ~]# usermod -G usersgrp user3
[root@localhost ~]# id user1
uid=1003(user1) gid=1006(usersgrp) groups=1006(usersgrp)
[root@localhost ~]# id user2
uid=1004(user2) gid=1006(usersgrp) groups=1006(usersgrp)
[root@localhost ~]# id user3
uid=1005(user3) gid=1005(user3) groups=1005(user3),1006(usersgrp)
[root@localhost ~]# cat /etc/group
root:x:0:
bin:x:1:
daemon:x:2:
sys:x:3:
adm:x:4:
tty:x:5:

dovenull:x:374:
sshd:x:74:
oprofile:x:16:
tcpdump:x:72:
student:x:1000:student
abc:x:1001:
pqr:x:1002:
user1:x:1003:
user2:x:1004:
user3:x:1005:
usersgrp:x:1006:user3
```

Working with Permissions

chmod command is used to change file permissions.

There are two methods to change permission: Relative Method and Absolute Method.

Categories

a: all users.

u: user owner.

g: group owner.

o: other user.

+: indicates apply permission.

-: indicates denial of permission.

```
[root@localhost FP]# touch file1.txt
[root@localhost FP]# ls -l
total 0
-rw-r--r--. 1 root root 0 Aug 27 20:29 file1.txt
```

#chmod a-rwx file1.txt: removes read, write and execution permission for all the users.

#chmod a+x file1.txt: adds executable permission for all the users.

#chmod g+r file1.txt: adds read permission to group owners.

#chmod g+w file1.txt: adds write permission to group owners.

#chmod a+rwx file1.txt: adds read, write and executable permissions for all the users.

```
[root@localhost FP]# chmod a-rwx file1.txt
[root@localhost FP]# ls -l
total 0
----------. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod a+x file1.txt
[root@localhost FP]# ls -l
total 0
---x--x--x. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod g+r file1.txt
[root@localhost FP]# ls -l
total 0
---xr-x--x. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod g+w file1.txt
[root@localhost FP]# ls -l
total 0
---xrwx--x. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod a+rwx file1.txt
[root@localhost FP]# ls -l
total 0
-rwxrwxrwx. 1 root root 0 Aug 27 20:37 file1.txt
```

#chmod g-r file1.txt: removes the read permission of group owner.

#chmod g-x file1.txt: removes the executable permission of group owner.

#chmod o-r file1.txt: removes the read permission of other users.

#chmod o-w file1.txt: removes the write permission of other users.

#chmod u-r file1.txt: removes the read permission of owner user.

#chmod u+r file1.txt: adds the read permission of owner user.

```
[root@localhost FP]# chmod g-r file1.txt
[root@localhost FP]# ls -l
total 0
-rwx-wxrwx. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod g-x file1.txt
[root@localhost FP]# ls -l
total 0
-rwx-w-rwx. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod o-r file1.txt
[root@localhost FP]# ls -l
total 0
-rwx-w--wx. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod o-w file1.txt
[root@localhost FP]# ls -l
total 0
-rwx-w---x. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod u-r file1.txt
[root@localhost FP]# ls -l
total 0
--wx-w---x. 1 root root 0 Aug 27 20:37 file1.txt
[root@localhost FP]# chmod u+r file1.txt
[root@localhost FP]# ls -l
total 0
-rwx-w---x. 1 root root 0 Aug 27 20:37 file1.txt
```

#chmod 777 file1.txt: it is absolute method to set read, write and executable permission.

```
[root@localhost FP]# chmod 777 file1.txt
[root@localhost FP]# ls -l
total 0
-rwxrwxrwx. 1 root root 0 Aug 27 20:37 file1.txt
```

setfacl: sets, modifies or remove the File Access Control List (FACL) to regular files and directories.

Here,

-m: creates new rule.

-r: remove the rule.

-b: back to normal.

getfacl: displays the comment header, base FACL entries, if there are any, for each file that is specified.

```
[root@localhost FP]# setfacl -m u:student:rwx file1.txt
[root@localhost FP]# getfacl file1.txt
# file: file1.txt
# owner: root
# group: root
user::rwx
user:student:rwx
group::rwx
mask::rwx
other::rwx

[root@localhost FP]# setfacl -x u:student file1.txt
[root@localhost FP]# getfacl file1.txt
# file: file1.txt
# owner: root
# group: root
user::rwx
group::rwx
mask::rwx
other::rwx
```