

Lab 3

1. Using the `useradd` command, add accounts for the following users in your system:

`user1`, `user2`, `user3`, `user4`, `user5`, `user6` and `user7`. Remember to give each user a password.

```
test>sudo useradd user1
[sudo] password for nabila:
test>sudo useradd user2
test>sudo useradd user3
test>sudo useradd user4
test>sudo useradd user5
test>sudo useradd user6
test>sudo useradd user7
```

A terminal window titled 'nabila@localhost:~' with a search icon and a menu icon in the top right corner. The terminal shows the following commands and output:

```
test>sudo passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>sudo passwd user2
Changing password for user user2.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>sudo passwd user3
Changing password for user user3.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>sudo passwd user4
Changing password for user user4.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
test>sudo passwd user5
Changing password for user user5.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>sudo passwd user6
Changing password for user user6.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>sudo passwd user7
Changing password for user user7.
New password:
BAD PASSWORD: The password is a palindrome
Retype new password:
passwd: all authentication tokens updated successfully.
test>
```

2. Using the groupadd command, add the following groups to your system.

Group	GID
sales	10000
hr	10001
web	10002

```
nabila@localhost:~
test>sudo groupadd -g 10000 sales
test>sudo groupadd -g 10001 hr
test>sudo groupadd -g 10002 web
test>
```

Why should you set GID in this manner instead of allowing the system to set the GID by default?

Ans: If I didn't give group id it will give it same as user id who create group.

3. Using the usermod command to add user1 and user2 to the sales auxiliary group, user3 and user4 to the hr auxiliary group. User5 and user6 to web auxiliary group. And add user7 to all auxiliary groups

```
nabila@localhost:~  
test>sudo usermod -aG sales user1  
[sudo] password for nabila:  
test>sudo usermod -aG sales user2  
test>sudo usermod -aG hr user3  
test>sudo usermod -aG hr user4  
test>sudo usermod -aG web user5  
test>sudo usermod -aG web user6  
test>sudo usermod -aG web user7  
test>sudo usermod -aG hr user7  
test>sudo usermod -aG sales user7
```

4. Login as each user and use id command to verify that they are in the appropriate groups. How else might you verify this information?

```
user5@localhost:/home/nabila  
test>su user1  
Password:  
[user1@localhost nabila]$ id  
uid=1002(user1) gid=1002(user1) groups=1002(user1),10000(sales) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user1@localhost nabila]$ su user2  
Password:  
[user2@localhost nabila]$ id  
uid=1003(user2) gid=1003(user2) groups=1003(user2),10000(sales) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user2@localhost nabila]$ su user3  
Password:  
[user3@localhost nabila]$ id  
uid=1004(user3) gid=1004(user3) groups=1004(user3),10001(hr) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user3@localhost nabila]$ su user4  
Password:  
[user4@localhost nabila]$ id  
uid=1005(user4) gid=1005(user4) groups=1005(user4),10001(hr) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[user4@localhost nabila]$ su user5  
Password:  
[user5@localhost nabila]$ id  
uid=1006(user5) gid=1006(user5) groups=1006(user5),10002(web) context=unconfined
```

```
nabila@localhost:~  
test>sudo groups user1  
[sudo] password for nabila:  
user1 : user1 sales  
test>sudo groups user2  
user2 : user2 sales  
test>sudo groups user3  
user3 : user3 hr  
test>sudo groups user4  
user4 : user4 hr  
test>sudo groups user5  
user5 : user5 web  
test>sudo groups user6  
user6 : user6 web  
test>sudo groups user7  
user7 : user7 sales hr web  
test>
```

5. Create a directory called /depts with a sales, hr, and web directory within the /depts directory.

```
nabila@localhost:~  
test>sudo mkdir -p /depts/sales  
test>sudo mkdir /depts/hr  
test>sudo mkdir /depts/web
```

6. Using the chgrp command, set the group ownership of each directory to the group with the matching name

```
nabila@localhost:~  
test>sudo chgrp sales /depts/sales  
test>sudo chgrp hr /depts/hr  
test>sudo chgrp web /depts/web  
test>
```

7. Set the permissions on the /depts directory to 755, and each subdirectory to 770

```
nabila@localhost:~  
test>sudo chmod 755 /depts  
test>sudo chmod 770 /depts/web  
test>sudo chmod 770 /depts/hr  
test>sudo chmod 770 /depts/sales
```

8. Set the set-gid bit on each departmental directory

```
nabila@localhost:/depts  
test>cd /depts  
test>sudo chmod g+s sales  
test>sudo chmod g+s hr  
test>sudo chmod g+s web
```

9. Use the su command to switch to the user2 account and attempt the following commands:

touch /depts/sales/user2.txt

touch /depts/hr/ user2.txt

touch /depts/web/ user2.txt

Which of these commands succeeded and which failed? What is the group ownership of the files that were created?

```
test>su user2  
Password:  
[user2@localhost nabila]$ touch /depts/sales/user2.txt  
[user2@localhost nabila]$ touch /depts/hr/ user2.txt  
touch: setting times of '/depts/hr/': Permission denied  
touch: cannot touch 'user2.txt': Permission denied  
[user2@localhost nabila]$ touch /depts/web/ user2.txt  
touch: setting times of '/depts/web/': Permission denied  
touch: cannot touch 'user2.txt': Permission denied  
[user2@localhost nabila]$
```

10. Configure sudoers file to allow user3 and user4 to use /bin/mount and /bin/umount commands, while allowing user5 only to use fdisk command.
user3 ALL=(root) /bin/mount , /bin/unmount

```
nabila@localhost:/  
test>sudo visudo /etc/sudoers  
[sudo] password for nabila:
```

```
nabila@localhost:/ — sudo visudo /etc/sudoers  
## users or groups.  
##  
## This file must be edited with the 'visudo' command.  
  
## Host Aliases  
## Groups of machines. You may prefer to use hostnames (perhaps using  
## wildcards for entire domains) or IP addresses instead.  
# Host_Alias      FILESERVERS = fs1, fs2  
# Host_Alias      MAILSERVERS = smtp, smtp2  
  
## User Aliases  
## These aren't often necessary, as you can use regular groups  
## (ie, from files, LDAP, NIS, etc) in this file - just use %groupnames  
## rather than USERALIASES  
# User_Alias ADMINS = jsmith, mikem  
user3 ALL=(root) /bin/mount , /bin/umount  
user4 ALL=(root) /bin/mount , /bin/umount  
user5 ALL=(root) /sbin/fdisk
```

11. Login by user3 and try to unmount /boot.

```
test>su user3  
Password:  
[user3@localhost /]$ sudo umount /boot  
[sudo] password for user3:  
[user3@localhost /]$
```

12. Login by user4 and remount /boot. Also try to view the partition table using fdisk.

```
user4@localhost:/

[user3@localhost ~]$ su user4
Password:
[user4@localhost ~]$ sudo mount /boot

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for user4:
[user4@localhost ~]$
```

```
[user4@localhost ~]$ fdisk -l
fdisk: cannot open /dev/nvme0n1: Permission denied
fdisk: cannot open /dev/sr0: Permission denied
fdisk: cannot open /dev/sr1: Permission denied
fdisk: cannot open /dev/mapper/rhel-root: Permission denied
fdisk: cannot open /dev/mapper/rhel-swap: Permission denied
[user4@localhost ~]$
```

13. Create a directory with permissions `rw-rwx---`, grant a second group (sales) `r-x` permissions

```
nabila@localhost:/

test>sudo mkdir dir1
[sudo] password for nabila:
test>chmod 770 dir1
chmod: changing permissions of 'dir1': Operation not permitted
test>sudo chmod 770 dir1
test>setfacl -m g:sales:rx dir1
setfacl: dir1: Operation not permitted
test>sudo setfacl -m g:sales:rx dir1
test>ls -ld dir1
drwxrwx---+ 2 root root 6 Feb 28 16:15 dir1
test>
```

14. create a file on that directory and grant read and write to a second group (sales)

```
nabila@localhost:~  
test>cd dir1  
test>touch file1  
test>cd ..  
test>sudo setfacl -m g:sales:rw dir1/file1  
test>
```

15. set the the owning group as the owning group of any newly created file in that directory.

```
nabila@localhost:~  
test>sudo chmod g+s dir1  
test>ls -ld dir1  
drwxrws---+ 2 nabila nabila 19 Feb 28 16:28 dir1  
test>
```