1- what happens when you run the passwd command
The passwd command would run as the root user since its the owner
2- what happens when you run the chrgrp command
Will run the command as the current user without the owner's privilege because it has no setuid
3- makin

```
[root@www ~]# useradd alice
[root@www ~]# useradd bob
[root@www ~]# passwd alice
Changing password for user alice.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@www ~]# passwd bob
Changing password for user bob.
New password:
[root@www ~]# passwd bob
Changing password for user bob.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@www ~]# groupadd staff
```

```
[root@www ~]# chmod o+t shared_data/
[root@www ~]# ll
total 192
-rw-------. 1 root root   919 Dec 20 13:37 anaconda-ks.cfg
-rw-r--r--. 1 root ftp  22592 Jan 18 13:35 get-docker.sh
-rw-r--r--. 1 root root 70152 May 10  2022 libcgroup-0.41-19.el8.x86_64.rpm
-rw-r--r--. 1 root root 93664 May 10  2022 libcgroup-tools-0.41-19.el8.x86_64
drwxr-xr-t. 2 root root     6 Jan 31 13:58 shared_data
drwxr-xr-x. 2 root root     6 Jan 18 14:50 static
drwxr-xr-x. 2 root root     6 Jan 29 16:20 web
[root@www ~]# # sticky bit --> directories
[root@www ~]# # prevents users in the same group with the same file permissio
[root@www ~]# chmod rwx shared_data/
chmod: invalid mode: ‘rwx’
Try 'chmod --help' for more information.
[root@www ~]# chmod g=rwx shared_data/
[root@www ~]# ll
total 192
-rw-------. 1 root root   919 Dec 20 13:37 anaconda-ks.cfg
-rw-r--r--. 1 root ftp  22592 Jan 18 13:35 get-docker.sh
-rw-r--r--. 1 root root 70152 May 10  2022 libcgroup-0.41-19.el8.x86_64.rpm
-rw-r--r--. 1 root root 93664 May 10  2022 libcgroup-tools-0.41-19.el8.x86_64
drwxrwxr-t. 2 root root     6 Jan 31 13:58 shared_data
drwxr-xr-x. 2 root root     6 Jan 18 14:50 static
drwxr-xr-x. 2 root root     6 Jan 29 16:20 web
[root@www ~]# # alice created --> file bob cant delete it
[root@www ~]#
```

```
drwxrwxr-t.   2 alice staff    6 Jan 31 13:58 shared_data
```

```
[root@www shared_data]# touch filealice
[root@www shared_data]# touch filebob
[root@www shared_data]# chwon alice filealice
bash: chwon: command not found...
Similar command is: 'chown'
[root@www shared_data]# chown alice filealice
[root@www shared_data]# chown bob filebob
[root@www shared_data]# ll
total 0
-rw-r--r--. 1 alice root 0 Jan 31 14:08 filealice
-rw-r--r--. 1 bob   root 0 Jan 31 14:08 filebob
[root@www shared_data]# su
```

```
[root@www shared_data]# su alice
[alice@www shared_data]$ rm filebob
rm: remove write-protected regular empty file 'filebob'? y
[alice@www shared_data]$ ll
total 0
-rw-r--r--. 1 alice alice 0 Jan 31 14:08 filealice
[alice@www shared_data]$ cd ..
[alice@www /]$ cd shared_data/
[alice@www shared_data]$ su bob
Password:
[bob@www shared_data]$ ll
total 0
-rw-r--r--. 1 alice alice 0 Jan 31 14:08 filealice
[bob@www shared_data]$ rm filealice
rm: remove write-protected regular empty file 'filealice'? y
rm: cannot remove 'filealice': Operation not permitted
[bob@www shared_data]$
```

Because alice is the dir owner it can delete bob's files while bob cant delete alice's files although he has permissions in the directory

```
[root@www shared_data]# getent group staff
staff:x:1004:alice,bob
[root@www shared_data]#
```

```
[root@www shared_data]# ll -d ../shared_data/
drwxrwxr-t. 2 alice staff 23 Jan 31 14:11 ../shared_data/
[root@www shared_data]#
```

We can see that the group owner is staff and they both belong to it

4- create backup.sh in /usr/local/bin with setuid

```
[root@www /]# touch /usr/local/bin/backup.sh
[root@www /]# ll /usr/local/bin/backup.sh
-rw-r--r--. 1 root root 0 Jan 31 15:10 /usr/local/bin/backup.sh
[root@www /]# chmod ugo+x /usr/local/bin/backup.sh
[root@www /]# chmod ugo+s /usr/local/bin/backup.sh
[root@www /]# ll /usr/local/bin/backup.sh
-rwsr-sr-x. 1 root root 0 Jan 31 15:10 /usr/local/bin/backup.sh
[root@www /]# chmod g-s /usr/local/bin/backup.sh
[root@www /]# ll /usr/local/bin/backup.sh
-rwsr-xr-x. 1 root root 0 Jan 31 15:10 /usr/local/bin/backup.sh
[root@www /]#
```

Anyone that would execute the backup.sh would run as the privilege of the root user

5- create shared_team directory

```
[root@www home]# ll
total 4
drwx------.  4 alice   alice     92 Jan 31 14:09 alice
drwx------.  4 bob     bob       92 Jan 31 14:10 bob
drwx-----x.  5 mostafa mostafa  132 Dec 29 19:02 mostafa
drwxr-xr-x.  2 root    root       6 Jan 31 15:15 shared_team
drwx------. 18 vomato  vomato  4096 Jan 26 16:43 vomato
[root@www home]# chmod g+s shared_team/
[root@www home]# ll
total 4
drwx------.  4 alice   alice     92 Jan 31 14:09 alice
drwx------.  4 bob     bob       92 Jan 31 14:10 bob
drwx-----x.  5 mostafa mostafa  132 Dec 29 19:02 mostafa
drwxr-sr-x.  2 root    root       6 Jan 31 15:15 shared_team
drwx------. 18 vomato  vomato  4096 Jan 26 16:43 vomato
[root@www home]#
```

6- add the sticky bit

```
[root@www home]# chmod o+t shared_team/
[root@www home]# ll
total 4
drwx------.  4 alice   alice     92 Jan 31 14:09 alice
drwx------.  4 bob     bob       92 Jan 31 14:10 bob
drwx-----x.  5 mostafa mostafa  132 Dec 29 19:02 mostafa
drwxr-sr-t.  2 root    root       6 Jan 31 15:15 shared_team
drwx------. 18 vomato  vomato  4096 Jan 26 16:43 vomato
[root@www home]#
```

7- whats the difference between the traditional acls and normal permissions

The difference between the acls is that acls are more fine grained targetted against certain users assigning named permissions and named groups and they go by order after DAC and the ACLS Are MACS must access user , named user, group , named group, other permissions

8- create alice and bob in lab_acls and add in group developers

```
[root@www /]# mkdir lab_acls
[root@www /]# ls
afs  bin  boot  dev  etc  home  lab_acls  lib  lib64  mec
[root@www /]# cd lab_acls
[root@www lab_acls]# ls
[root@www lab_acls]# groupadd developers
[root@www lab_acls]# gpasswd -a alice developers
Adding user alice to group developers
[root@www lab_acls]# gpasswd -a bob developers
Adding user bob to group developers
[root@www lab_acls]# testfile.txt
bash: testfile.txt: command not found...
[root@www lab_acls]# touch testfile.txt
[root@www lab_acls]#
```

9/10-getfacl of the file testfile.txt

```
[root@www lab_acls]# setfacl -m "user:alice:rw" testfile.txt
[root@www lab_acls]# getfacl testfile.txt
# file: testfile.txt
# owner: root
# group: root
user::rw-
user:alice:rw-
group::r--
mask::rw-
other::r--
```

11- add execute for group developers

```
[root@www lab_acls]# getfacl testfile.txt
# file: testfile.txt
# owner: root
# group: root
user::rw-
user:alice:rw-
group::r--
group:developers:--x
mask::rwx
other::r--
```

12-remove the alice named permission

```
[root@www lab_acls]# setfacl -x "u:alice" testfile.txt
[root@www lab_acls]# getfacl testfile.txt
# file: testfile.txt
# owner: root
# group: root
user::rw-
group::r--
group:developers:--x
mask::r-x
other::r--
```

13- set default acl on the dir mydir

```
[root@www lab_acls]# mkdir mydir
[root@www lab_acls]# setfacl -dm "m:rwx" mydir
[root@www lab_acls]# getfacl mydir
# file: mydir
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:mask::rwx
default:other::r-x

[root@www lab_acls]#
```

14- set default acl on mydir for group developers to be read and write

```
[root@www lab_acls]# setfacl -dm "g:developers:rw" mydir
[root@www lab_acls]# getfacl mydir
# file: mydir
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:developers:rw-
default:mask::rwx
default:other::r-x

[root@www lab_acls]#
```

15- set privelege of bob to be read and execute

```
[root@www lab_acls]# setfacl -dm "user:bob:rx" .
[root@www lab_acls]# getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:bob:r-x
group::r-x
mask::r-x
other::r-x
default:user::rwx
default:user:bob:r-x
default:group::r-x
default:mask::r-x
default:other::r-x
```

16- how does the mask affect permissions

The mask is bitwise anded to the permissions given to make the effective mask to be the intersection between the given permission and the mask

17- setting the mask to r–

```
[root@www lab_acls]# setfacl -m "m:r--" .
[root@www lab_acls]# getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:bob:r-x                    #effective:r--
group::r-x                      #effective:r--
mask::r--
other::r-x
default:user::rwx
default:user:bob:r-x
default:group::r-x
default:mask::r-x
default:other::r-x
```

18- adding rw for alice and bob in testfile.txt in one commadn

```
[root@www lab_acls]# setfacl -m "user:alice:rw,user:bob:rw" testfile.txt
[root@www lab_acls]# getfacl testfile.txt
# file: testfile.txt
# owner: root
# group: root
user::rw-
user:alice:rw-
user:bob:rw-
group::r--
group:developers:--x
mask::rwx
other::r--

[root@www lab_acls]#
```

19- backup the acls

```
[root@www lab_acls]# getfacl mydir > mydir_acls.txt
[root@www lab_acls]# ll
total 4
drwxr-xr-x+ 2 root root   6 Jan 31 15:44 mydir
-rw-r--r--+ 1 root root 178 Jan 31 15:57 mydir_acls.txt
-rw-rwxr--+ 1 root root   0 Jan 31 15:33 testfile.txt
[root@www lab_acls]# cat mydir_acls.txt
# file: mydir
# owner: root
# group: root
user::rwx
group::r-x
other::r-x
default:user::rwx
default:group::r-x
default:group:developers:rw-
default:mask::rwx
default:other::r-x

[root@www lab_acls]#
```

20- add shared

Rw access by developers

```
[root@www /]# mkdir shared
[root@www /]# setfacl -m "group:developers:rw" shared
[root@www /]# chgrp developers shared
[root@www /]# ll
total 28
```

```
[root@www /]# chmod g+s developers
chmod: cannot access 'developers': No such file or directory
[root@www /]# chmod g+s shared
[root@www /]# chmod +t shared
[root@www /]# ll
total 28
dr-xr-xr-x.   2 root root          6 Jun 25  2024 afs
lrwxrwxrwx.   1 root root          7 Jun 25  2024 bin -> usr/bin
dr-xr-xr-x.   5 root root       4096 Jan 26 23:04 boot
drwxr-xr-x.  20 root root       3440 Jan 31 13:41 dev
drwxr-xr-x. 140 root root       8192 Jan 31 15:33 etc
drwxr-xr-x.   7 root root         78 Jan 31 15:15 home
drwxrwxr-x+   3 root root         61 Jan 31 15:57 lab_acls
lrwxrwxrwx.   1 root root          7 Jun 25  2024 lib -> usr/lib
lrwxrwxrwx.   1 root root          9 Jun 25  2024 lib64 -> usr/lib64
drwxr-xr-x.   2 root root          6 Jun 25  2024 media
drwxr-xr-x.   3 root root         18 Jun 25  2024 mnt
drwxr-xr-x.   3 root root         24 Jun 25  2024 opt
dr-xr-xr-x. 333 root root          0 Jan 31 13:41 proc
dr-xr-x---.   7 root root       4096 Jan 31 15:42 root
drwxr-xr-x.  49 root root       1360 Jan 31 13:41 run
lrwxrwxrwx.   1 root root          8 Jun 25  2024 sbin -> usr/sbin
drwxrwsr-t+   2 root developers    6 Jan 31 16:01 shared
drwxrwxr-t.   2 root staff        38 Jan 31 14:44 shared_data
```