how to set sticky bit

In our previous articles we have discussed about read write and execute permission for file and directory. Now I will show you some special permission which you can set for files and directories.

Ownership issue

In some case you want to grant permission to other user while keeping ownership to self.
s permission
is used to deal with this situations. s options is used to add both user ID and group ID permission to a file.
The following example add's user ID permission to the pppd program, which is owned by the root user. When an ordinary user runs pppd, the root user retains ownership, allowing the pppd program to change root-owned files.

```
 # chmod +s /usr/sbin/pppd
```

The Set User ID and Set Group ID permissions show up as an s in the execute position of the owner and group segments. Set User ID and Group ID are essentially variations of the execute permission, x. Read, write, and User ID permission are rws instead of just rwx.

```
# ls -l
/usr/sbin/pppd -rwsr-sr-x 1 root root 18666 Jan 12 12:48 /usr/sbin/pppd
```

Sticky Bit Permissions

Sticky Bit is used for directories to protect files within them. Files in a directory with the sticky bit set can only be deleted or renamed by the root user or the owner of the directory.

Sticky Bit Permission Using Symbols

The sticky bit permission symbol is **t**. The sticky bit shows up as a t in the execute position of the other permissions. A program with read and execute permissions with the sticky bit has its permissions displayed as r-t.

```
# chmod +t /home/vinita/data
# ls -l /home/vinita/data -rwxr-xr-t 1 root root 4096 /home/vinita/data
```

Sticky Bit Permission Using the Binary Method

As with ownership, for sticky bit permissions, you add another octal number to the beginning of the octal digits. The octal digit for the sticky bit is 1 (001). The following example sets the sticky bit for the data directory:

```
# chmod 1755 /home/vinita/data
```

The next example sets both the sticky bit and the User ID permission on the newprogs directory.

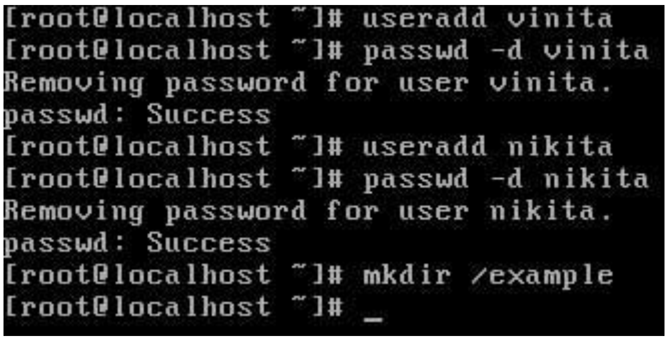
```
# chmod 5755 /usr/bin/newprogs
# ls -l /usr/bin/newprogs drwsr-xr-t 1 root root 4096 /usr/bin/newprogs
```

Sticky bit example of practically implementations

USER ID and GROUP ID Permissions

To understand sticky bit and user permission in more depth let's take an example. Create two user named vinita and nikita. And a example directory on root partitions.

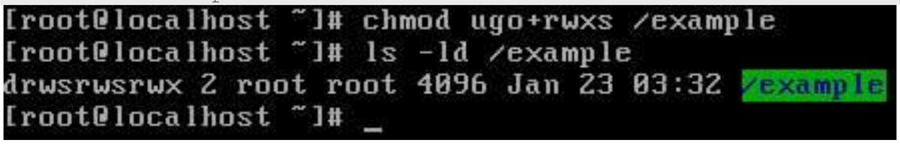
```
#useradd vinita
#passwd -d vinita
#useradd nikita
#passwd -d nikita
#mkdir /example
```



```
[root@localhost ~]# useradd vinita
[root@localhost ~]# passwd -d vinita
Removing password for user vinita.
passwd: Success
[root@localhost ~]# useradd nikita
[root@localhost ~]# passwd -d nikita
Removing password for user nikita.
passwd: Success
[root@localhost ~]# mkdir /example
[root@localhost ~]# _
```

As example directory is created by root so the owner and group of this directory will root. By default permission will be inherited to all other object created in this directory to root owner. Now we will use symbolic method to change the ownership issue to this directory.

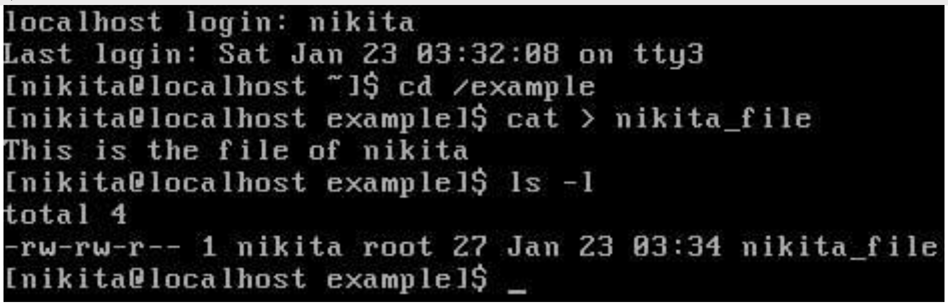
```
#chmod ugo+rwxs /example
#ls -ld /example
```



```
[root@localhost ~]# chmod ugo+rwxs /example
[root@localhost ~]# ls -ld /example
drwsrwsrwx 2 root root 4096 Jan 23 03:32 /example
[root@localhost ~]# _
```

As you can see in image s bit is set in owner and group filed which will automatically set owner and group to their respective owner and group. To verify login form user nikita and change directory to example and creates a file.

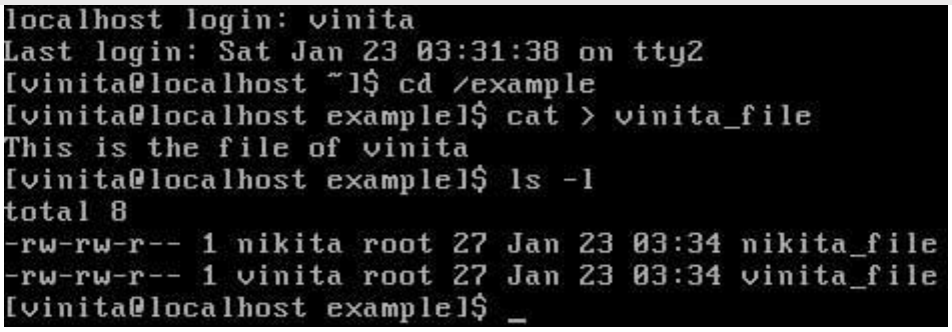
```
$cd /example
$cat > nikita_file
This is the file of nikita
$ls -l
```



```
localhost login: nikita
Last login: Sat Jan 23 03:32:08 on tty3
[nikita@localhost ~]$ cd /example
[nikita@localhost example]$ cat > nikita_file
This is the file of nikita
[nikita@localhost example]$ ls -l
total 4
-rw-rw-r-- 1 nikita root 27 Jan 23 03:34 nikita_file
[nikita@localhost example]$ _
```

As you can see owner filed is changed to user nikita.
Now create a file form user vinita.

```
$cd /example
$cat > vinita_file
This is file of Vinita
$ls -ld
```

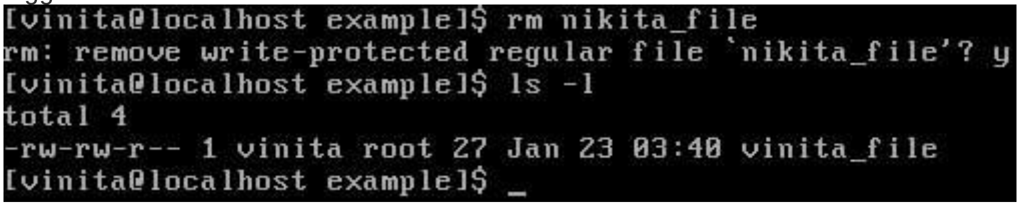


```
localhost login: vinita
Last login: Sat Jan 23 03:31:38 on tty2
[vinita@localhost ~]$ cd /example
[vinita@localhost example]$ cat > vinita_file
This is the file of vinita
[vinita@localhost example]$ ls -l
total 8
-rw-rw-r-- 1 nikita root 27 Jan 23 03:34 nikita_file
-rw-rw-r-- 1 vinita root 27 Jan 23 03:34 vinita_file
[vinita@localhost example]$ _
```

Now you can understand what s bit do in chmod command. This is best options when users are working on some shared project. As they will gets ownership of their files automatically.

Implementation of sticky bit

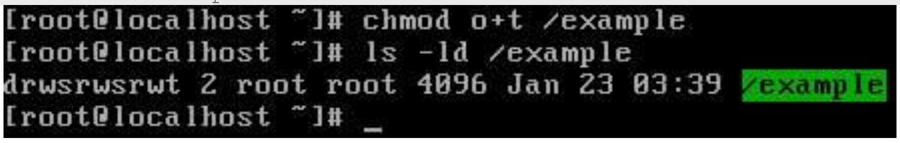
But this could create other problem. User can accidentally or intensely delete other user's files and folder as all user have full permission on this shared folder. Go on terminal where user Vinita is logged in and delete the file of nikita.



```
[vinita@localhost example]$ rm nikita_file
rm: remove write-protected regular file `nikita_file'? y
[vinita@localhost example]$ ls -l
total 4
-rw-rw-r-- 1 vinita root 27 Jan 23 03:40 vinita_file
[vinita@localhost example]$ _
```

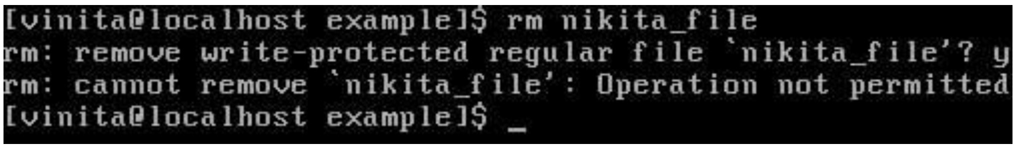
To control this behaviors switch to root user and set sticky bit on /example folder.

```
#chmod o+t /example
#ls -ld /example
```



```
[root@localhost ~]# chmod o+t /example
[root@localhost ~]# ls -ld /example
drwsrwsrwt 2 root root 4096 Jan 23 03:39 /example
[root@localhost ~]# _
```

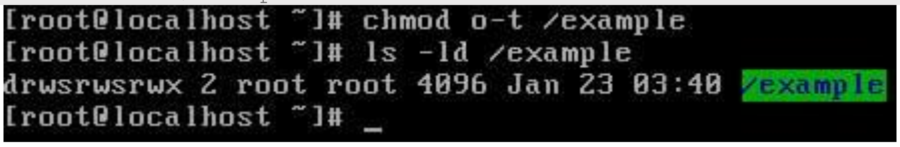
Sticky bit is defined by t options. As you can see in output other have t bit set in their filed. Now only owner of file and root user can delete file in this folder.
To verify switch Vinita user again and try to delete the files of nikita. This time it will not success this time.



```
[vinita@localhost example]$ rm nikita_file
rm: remove write-protected regular file `nikita_file'? y
rm: cannot remove `nikita_file': Operation not permitted
[vinita@localhost example]$ _
```

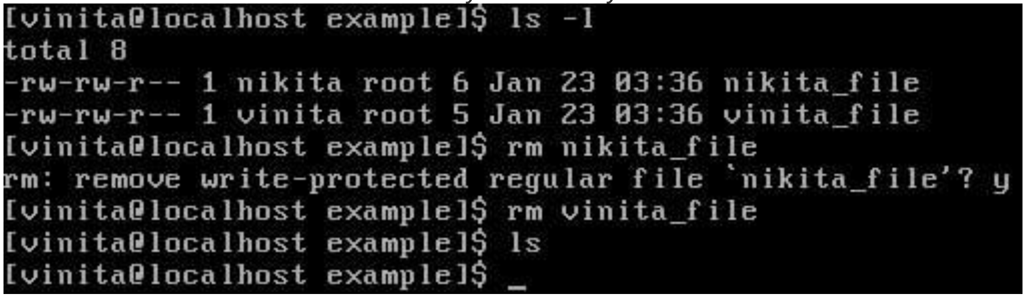
To remove sticky bit use minus sign.

```
#chmod o-t /example
```



```
[root@localhost ~]# chmod o-t /example
[root@localhost ~]# ls -ld /example
drwsrwsrwx 2 root root 4096 Jan 23 03:40 /example
[root@localhost ~]# _
```

now Vinita can delete the files owned by nikita verify



```
[vinita@localhost example]$ ls -l
total 8
-rw-rw-r-- 1 nikita root 6 Jan 23 03:36 nikita_file
-rw-rw-r-- 1 vinita root 5 Jan 23 03:36 vinita_file
[vinita@localhost example]$ rm nikita_file
rm: remove write-protected regular file `nikita_file'? y
[vinita@localhost example]$ rm vinita_file
[vinita@localhost example]$ ls
[vinita@localhost example]$ _
```