BASIC FILE ATTRIBUTES

CONTENTS

- ls –l to display file attributes (properties)
- Listing of a specific directory
- Ownership and group ownership
- Different file permissions

- A file has number of attributes(properties) that are stored in the inode
- **ls** command looks up the file's inode to fetch its attributes
- **ls -l** to list seven attributes of all files in the current directory

LISTING FILE ATTRIBUTES

ls –l provides attributes like

- File type and Permissions
- Links: number of links associated with file.This is actually the number of filenames maintained by the system of that file
- Owner (Ownership):when you create a file,you automically become the owner(create,delete or modify)
- Group owner(Group Ownership):when opening a user account, the system administrator also assigns the user to some group

- Size:size of file in bytes
- Last modification time
- File name: displays the filenames arranged in ASCII collating sequence. Unix filenames can be very long (upto 255 characters).If you would like to see an important file at the top of the listing,then choose its name in uppercase –atleast, its first letter

- $ ls –l

```
total 72
-rw-r--r--   1  kumar  metal  19514  may  10  13:45 chap01
-rw-r--r--   1  kumar  metal   4174  may  10  15:01 chap02
-rw-rw-rw-   1  kumar  metal     84  feb  12  12:30 dept.lst
-rw-r--r--   1  kumar  metal   9156  mar  12  1999 genie.sh
drwxr-xr-x   2  kumar  metal    512  may   9  10:31 helpdir
drwxr-xr-x   2  kumar  metal    512  may   9  09:57 progs
```

Total 72, which indicates that a total of 72 blocks are occupied by these files on disk, each block consisting of 512 bytes(1024 in linux)

1. The file type and permissions associated with each file
2. The number of file names maintained by the system. This does not mean that there are two copies of the file
3. File created by the owner
4. Every user is attached to a group owner

5. File size in bytes
6. Last modification time. If you change only the permissions or ownership of the file, the modification time remains unchanged
7. Displays file name

LISTING DIRECTORY ATTRIBUTES

- ls –ld helpdir progs

```
drwxr-xr-x  2  kumar  metal      512  may   9   10:31 helpdir
drwxr-xr-x  2  kumar  metal      512  may   9   09:57 progs
```

Note: ls -d will not list all subdirectories in the current directory

FILE OWNERSHIP

- When you create a file, you become its owner (third column)
- Group owner of the file (fourth column)
- Several users may belong to a single group, but **the privileges of the group are set by the owner of the file and not by the group members**

- When the system administrator creates a user account,he has to assign these parameters to the user:

->**The user-id (UID)--both its name and numeric representation**

->The group-id (GID)--both the name and numeric representation

- The file /etc/passwd maintains theUID(both number and name) and GID(but only the number)
- /etc/group contains the GID(both number and name)

- To know your own UID and GID without viewing /etc/passwd and /etc/group, use the id command

```
$ id
uid =655537 (kumar)
gid =655535 (metal)
```

FILE PERMISSIONS

- UNIX follows a three-tiered file protection system that determines a file's access rights
- Filetype owner (rwx) groupowner (rwx) others (rwx)
- Example:

-rwxr-xr-- 1 kumar metal 20500 may 10 19:21 chap02

r w x	**r - x**	**r - -**
owner/user	group owner	others

- Each group here represents a category and contains three slots, representing the read, write and execute permission of the file-

r indicates read permission,which means cat can display the file

w indicates write permission; you can edit such a file with editor

x indicates execute permission;the file can be executed as a permission

- absence of the corresponding permission

CHANGING FILE PERMISSIONS

- A file or a directory is created with a default set of permissions, which can be determined by umask
- Let us assume that the file permission for the created file is -rw-r--r--
- Using **chmod(change mode)** command: used to set the permission of one or more files for all categories of users(user, group and others).It can be run only by the user(the owner) and the superuser.

RELATIVE AND ABSOLUTE PERMISSIONS

- In a relative manner, specify the changes to the current permissions

- In an absolute manner, specify the final permissions

RELATIVE PERMISSIONS

- chmod only changes the permissions specified in the command line and leaves the other permissions unchanged
- Syntax

  chmod category operation permission filename(s)

- chmod takes an expression as its argument which contains:

1. user category (user, group, others)
2. operation to be performed (assign or remove a permission)
3. type of permission (read, write, execute)

Category	**operation**	**permission**
u - user	+ assign	r - read
g - group	- remove	w - write
o - others	= absolute	x - execute
a - all (ugo)		

- Examples

```
-rw-r--r-- 1  kumar     metal  1906   sep     23:38 xstart
```

```
chmod u+x xstart
```

```
-rwxr--r-- 1  kumar     metal  1906   sep     23:38 xstart
```

The command assigns (+) execute (x) permission to the user (u), other permissions remain unchanged

- chmod ugo+x xstart
- chmod a+x xstart //a implies ugo
- chmod +x xstart //default, a is implied

```
-rwxr-xr-x 1 kumar  metal  1906   sep
    23:38   xstart
```

chmod accepts multiple file names in command line

- chmod u+x note note1 note3

```
$chmod go-r xstart
```

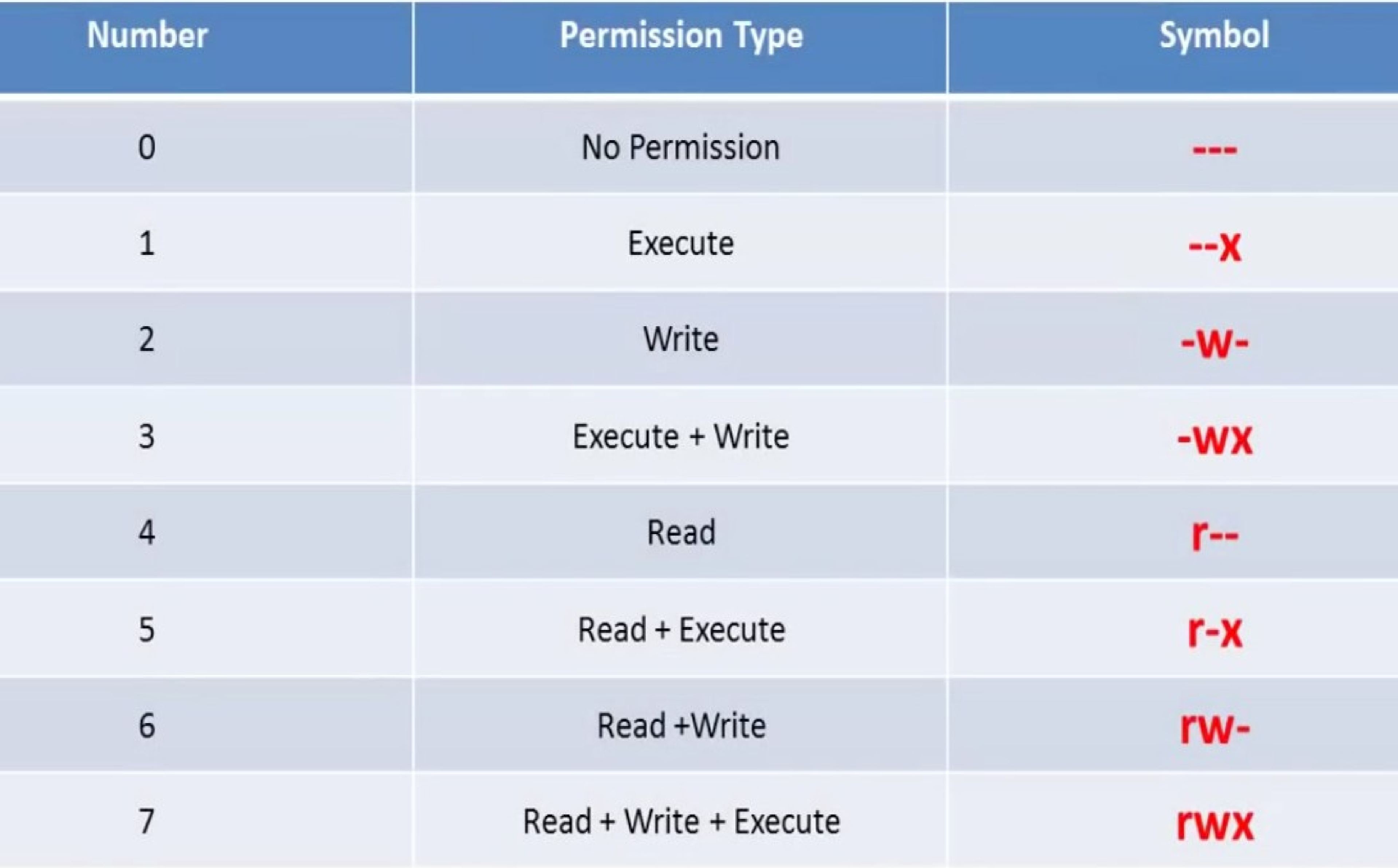
Let initially

```
-rwxr-xr-x 1   kumar   metal   1906   sep
   23:38    xstart
```

Then, it becomes

```
-rwx--x--x 1   kumar   metal   1906   sep
   23:38    xstart
```

Absolute Permission



Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write + Execute	rwx

```
$chmod 642 xstart
```

Using chmod recursively(-R)

- chmod descend a directory hierarchy and apply the expression to every file and subdirectory it finds.This is done with -R(recursive) option:

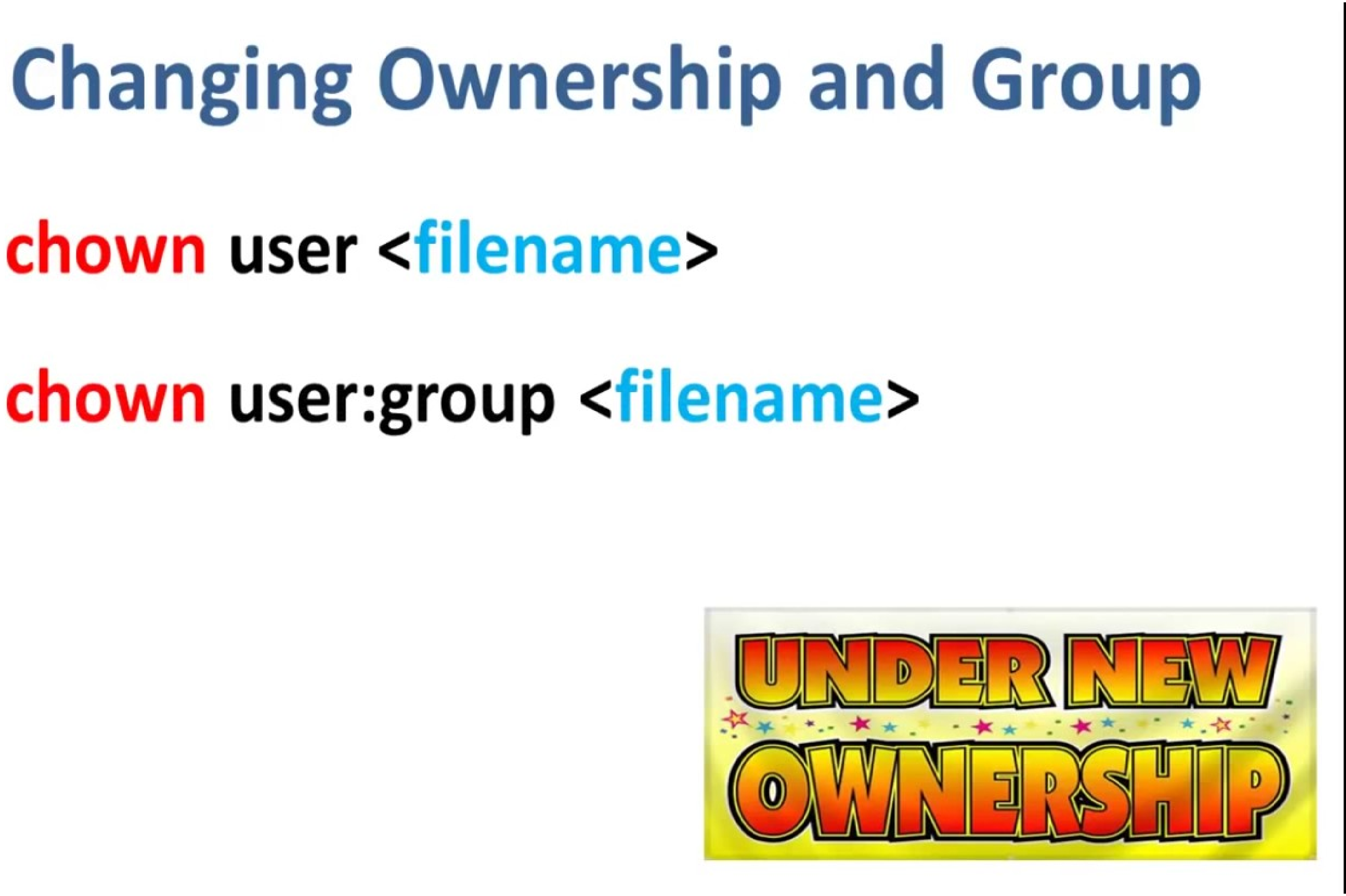
chmod -R 755 .   //works on hidden files also

chmod -R a+x *   //leaves out hidden files

Changing Ownership

- Changing file owner(chown)
- Changing group owner(chgrp)

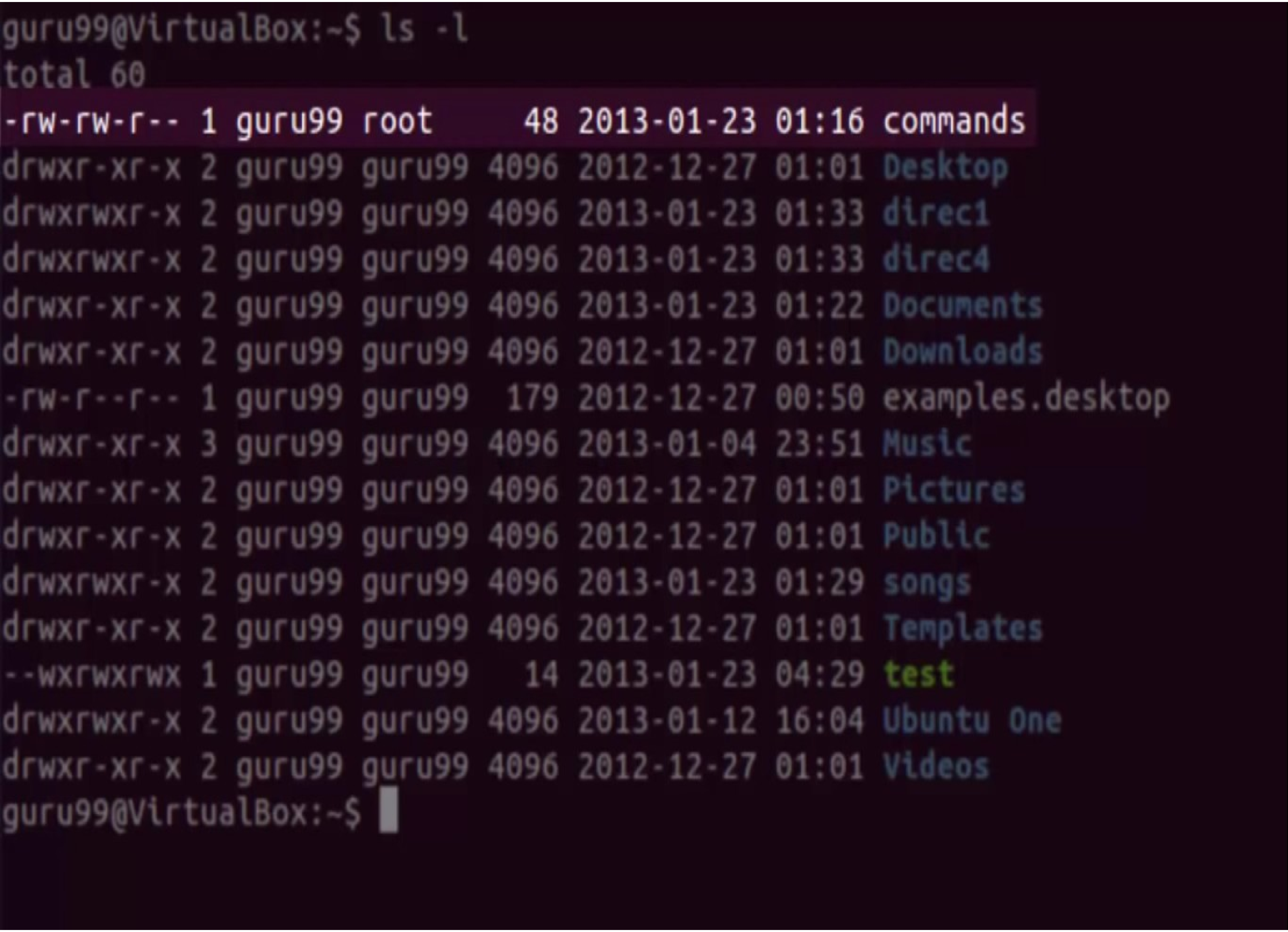
chown



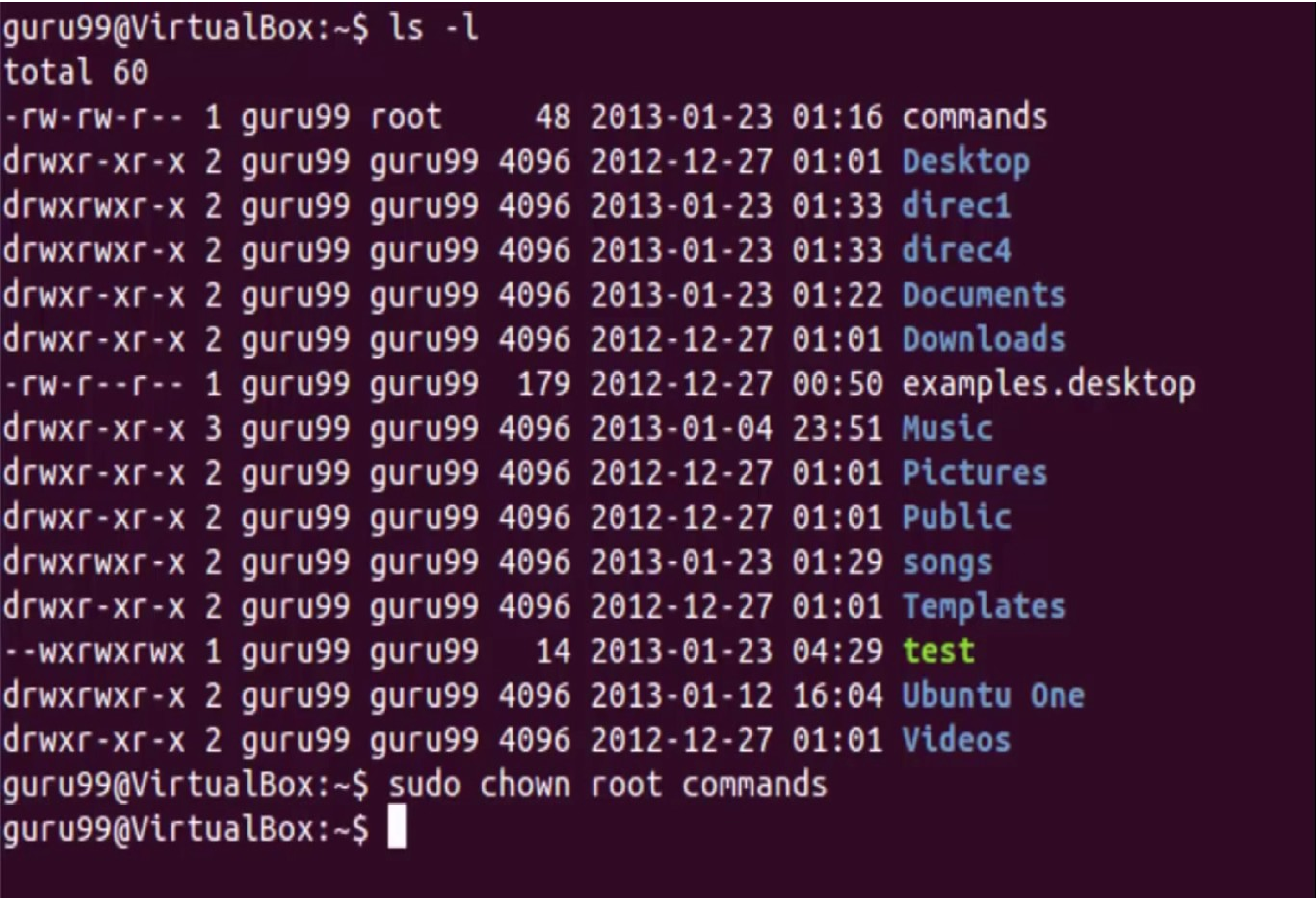
Changing Ownership and Group

```
chown user <filename>

chown user:group <filename>
```



```
guru99@VirtualBox:~$ ls -l
total 60
-rw-rw-r-- 1 guru99 root     48 2013-01-23 01:16 commands
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Desktop
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:33 direc1
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:33 direc4
drwxr-xr-x 2 guru99 guru99 4096 2013-01-23 01:22 Documents
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Downloads
-rw-r--r-- 1 guru99 guru99  179 2012-12-27 00:50 examples.desktop
drwxr-xr-x 3 guru99 guru99 4096 2013-01-04 23:51 Music
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Pictures
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Public
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:29 songs
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Templates
--wxrwxrwx 1 guru99 guru99   14 2013-01-23 04:29 test
drwxrwxr-x 2 guru99 guru99 4096 2013-01-12 16:04 Ubuntu One
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Videos
guru99@VirtualBox:~$ 
```



```
guru99@VirtualBox:~$ ls -l
total 60
-rw-rw-r-- 1 guru99 root     48 2013-01-23 01:16 commands
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Desktop
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:33 direc1
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:33 direc4
drwxr-xr-x 2 guru99 guru99 4096 2013-01-23 01:22 Documents
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Downloads
-rw-r--r-- 1 guru99 guru99  179 2012-12-27 00:50 examples.desktop
drwxr-xr-x 3 guru99 guru99 4096 2013-01-04 23:51 Music
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Pictures
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Public
drwxrwxr-x 2 guru99 guru99 4096 2013-01-23 01:29 songs
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Templates
--wxrwxrwx 1 guru99 guru99   14 2013-01-23 04:29 test
drwxrwxr-x 2 guru99 guru99 4096 2013-01-12 16:04 Ubuntu One
drwxr-xr-x 2 guru99 guru99 4096 2012-12-27 01:01 Videos
guru99@VirtualBox:~$ sudo chown root commands
guru99@VirtualBox:~$ 
```

```
$ su
Password:                    //This is root password
#_                           //This is another shell

# ls -l note
-rwxr----x 1 kumar metal 347 May 10 20:30
   note
#chown sharma note;ls -l note
-rwxr----x 1 sharma metal 347 May 10 20:30
note
```

chgrp:changing group owner

- Shares a similar syntax with chown