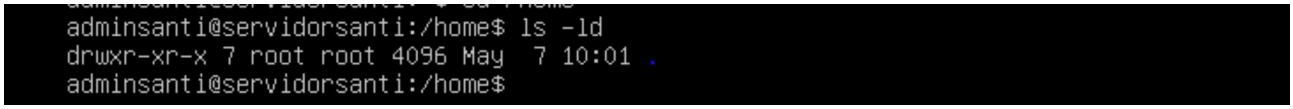
PRACTICE 4. Managing Files and Directories.

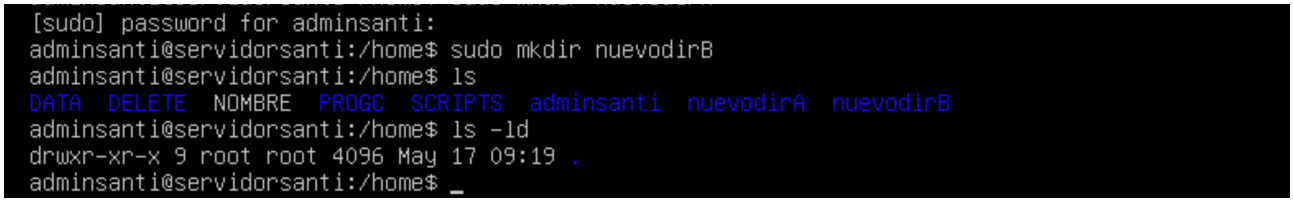
1. Obtain the number of hard links in the HOME directory.

First, to understand this, let's talk about inodes. The data in the filesystem is held in blocks on the disk, and those blocks are collected together by an inode. You can think of the inode as THE file. Inodes lack filenames, though. That's where links come in.
A link is just a pointer to an inode. A directory is an inode that holds links. Each filename in a directory is just a link to an inode. Opening a file in Unix also creates a link, but it's a different type of link (it's not a named link). (https://unix.stackexchange.com/questions/22394/why-are-hard-links-to-directories-not-allowed-in-unix-linux=)



2. Create two subdirectories of HOME and repeat the previous point. What difference there are? explain what happened.

Since two extra subdirectories have been created there has to be two new hardlinks that point the inodes of this subdirectories.



3. Create a nonsymbolic link (hard) and a symbolic link on a file. Obtain theinode numbers of the three files (the two links and the original) and compare them. Try to do the same for a directory.

The hard link have the same inode number

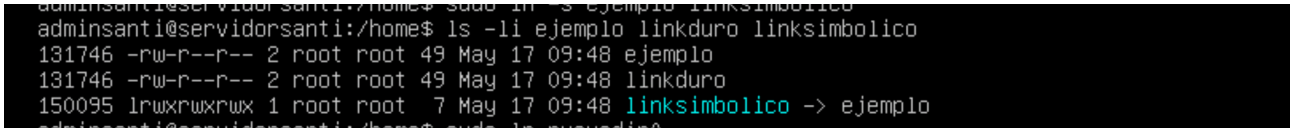
Sudo ln ejemplo linkduro
Sudo ln -s ejemplo linksimbolico

```
adminsanti@servidorsanti:/home$ ls -li ejemplo linkduro linksimbolico
131746 -rw-r--r-- 2 root root 49 May 17 09:48 ejemplo
131746 -rw-r--r-- 2 root root 49 May 17 09:48 linkduro
150095 lrwxrwxrwx 1 root root  7 May 17 09:48 linksimbolico -> ejemplo
adminsanti@servidorsanti:/home$ _
```

The hard link is not allowed. The symbolic link is allowed

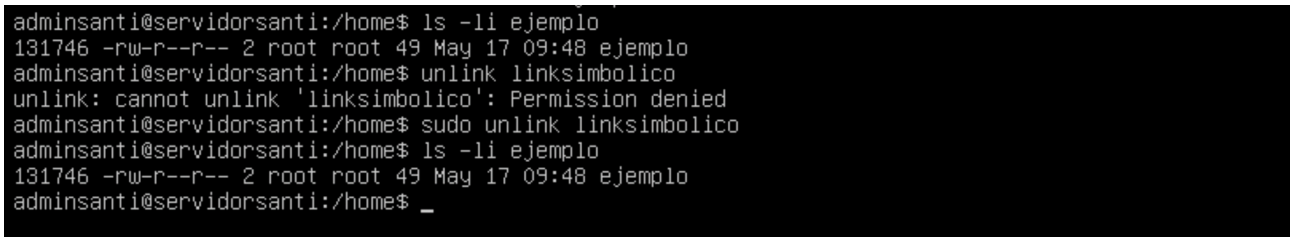
```
adminsanti@servidorsanti:/home$ sudo ln nuevodirA linkduronuevodirA
ln: nuevodirA: hard link not allowed for directory
adminsanti@servidorsanti:/home$ sudo ln -s nuevodirA linksimboliconuevodirA
adminsanti@servidorsanti:/home$ ls
DATA    NOMBRE  SCRIPTS     ejemplo   linksimbolico          nuevodirA
DELETE  PROGC   adminsanti  linkduro  linksimboliconuevodirA  nuevodirB
adminsanti@servidorsanti:/home$
```

4. Obtain a list ls -l of the directory where you created the ligatures. What are the differences in the file type of these three files?



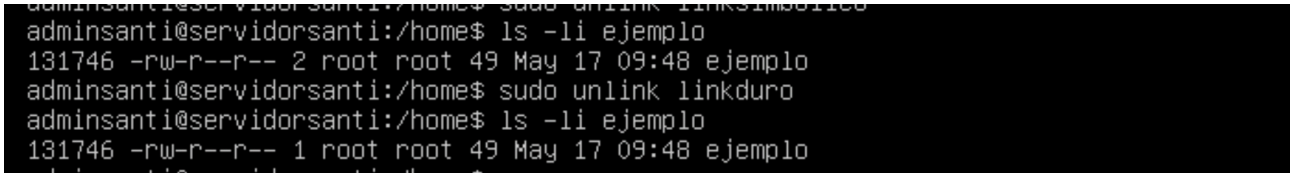
The firt two are regular files (-) but the last one is a symbolic link (l)

5. **Contabilize the number of ligatures of the previous original file. Delete the soft ligature and recount the number of ligatures.**



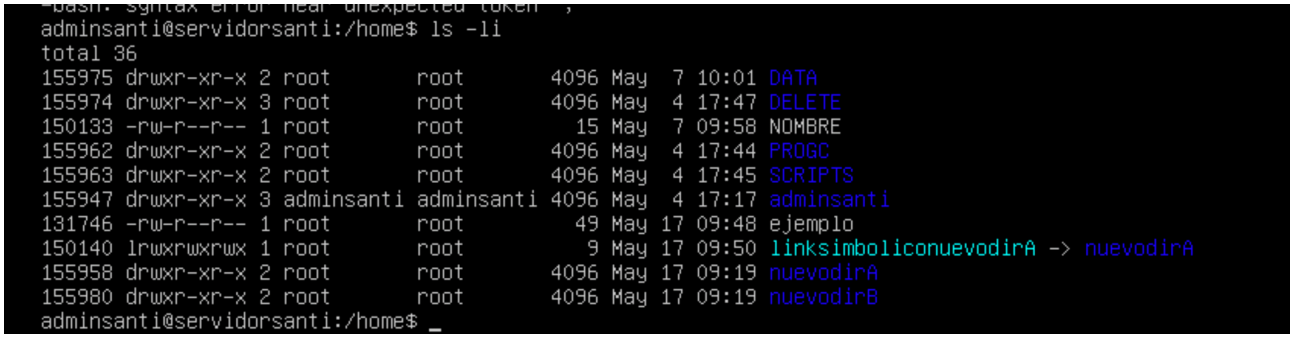
6. Delete the ligature lasts and re-counts the number of ligatures. What is the difference between what we have obtained in point 4 and 5?

The number of ligatures now is 1.



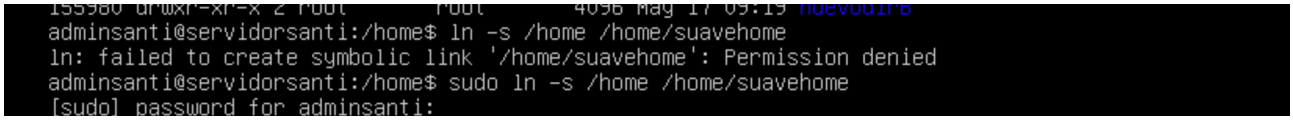
7. Create a soft link to the HOME directory.

I’m gonna check the original ligatures for /home



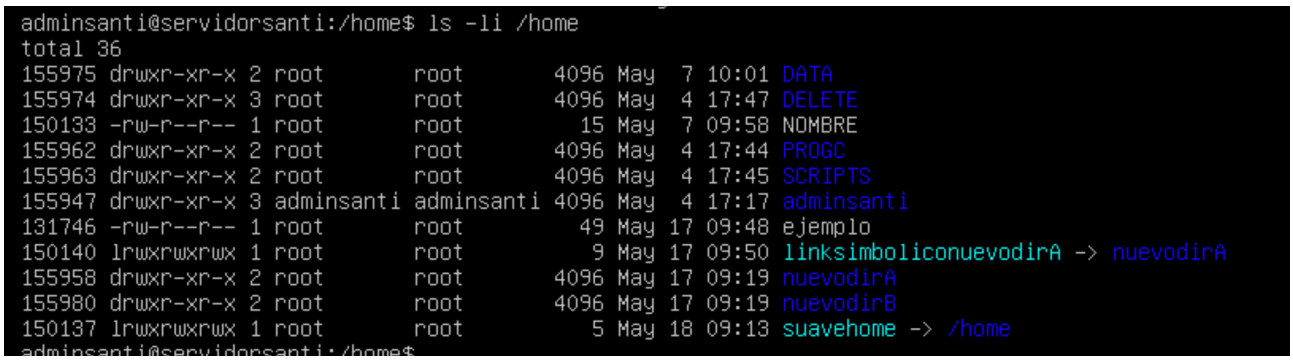
THEN:

Using ln -s /home /home/suavehome



- **Has the number of ligatures changed?**

Yes, now there is one mor (before: 19, now: 20)



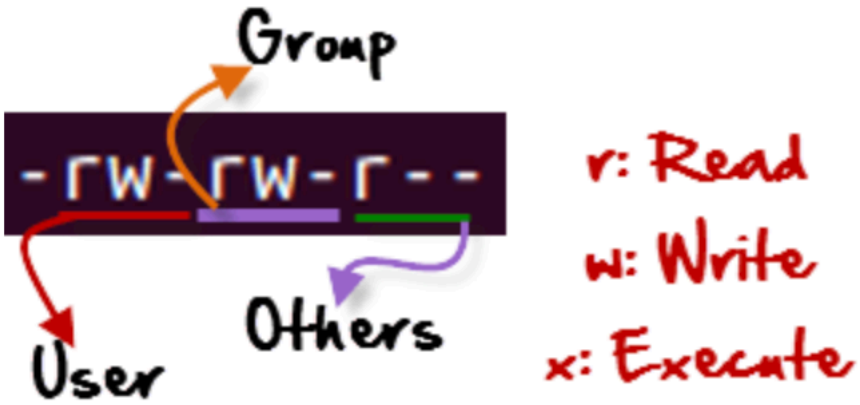
- **Change to the newly created directory. In what directory are we really?**

  We are still at home directory.

8. Observe the access rights of the different files in the dir HOME.

Para ". " y para ".." los permisos son drwxr-xr-x
Para los directorios : drwxr-xr-x
Para los cat : -rw-r—r—

The fiirst symbol is the type of file:
r = read permission
w = write permission
x = execute permission
- = no permission
¿Cómo leer el resto de bits?
Van agrupados de tres en tres:
-los tres primeros para el usuario
-los tres siguiuentes para el grupo
-los ultimos para el "mundo"[1]



[1] Guru99. "File Permissions in Linux/Unix: How to Read/Write & Change?"[en línea]. *Guru99.* https://www.guru99.com/file-permissions.html#:~:text=There%20are%20three%20user%20types,into%20Absolute%20and%20Symbolic%20mode [consultada: 18 mayo 2021].

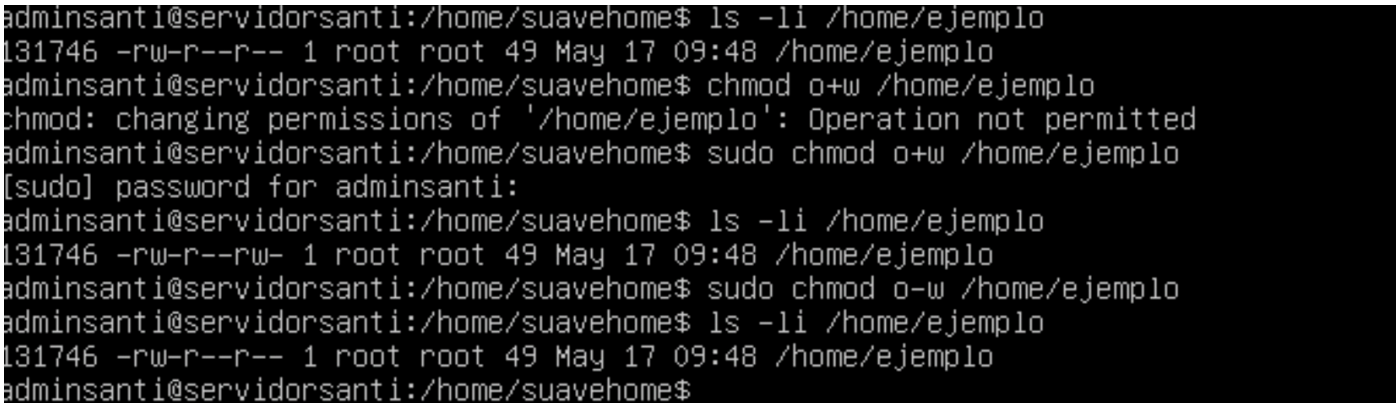
9. Establishes and removes rights on a file to other users

Using the comand chmod. In the example I added write permission and the deleted it

Operator	Description
+	Adds a permission to a file or directory
-	Removes the permission
=	Sets the permission and overrides the permissions set earlier.

The various owners are represented as -

User Denotations	
u	user/owner
g	group
o	other
a	all



10. Change the owner of a file.

In the above example the owner is “root” (ls -l shows it). I used chown to change it:

```
adminsanti@servidorsanti:/home/suavehome$ ls -l /home/ejemplo
-rw-r--r-- 1 root root 49 May 17 09:48 /home/ejemplo
adminsanti@servidorsanti:/home/suavehome$ sudo chown adminsanti /home/ejemplo
adminsanti@servidorsanti:/home/suavehome$ ls -l /home/ejemplo
-rw-r--r-- 1 adminsanti root 49 May 17 09:48 /home/ejemplo
adminsanti@servidorsanti:/home/suavehome$
```

11. Give complete permissions in the files that the user creates and make sure that the others can only read and execute them.

Por defecto los usuarios tendran estos permisos si uso umask[2]

Use `umask` followed by the mask representing what you want to **deny**.

Command	Filters	Description	Created directories will have this permission	Created files will have this permission
`$ umask 000`	`----------`	Deny nobody anything.	`drwxrwxrwx`	`rw-rw-rw-`
`$ umask 006`	`------rw-`	Deny rw to others, but allow everyone to list directories	`drwxrwx--x`	`rw-rw----`
`$ umask 007`	`------rwx`	Deny rwx to others	`drwxrwx---`	`rw-rw----`
`$ umask 077`	`---rwxrwx`	Deny rwx to others and to the group. Only you can access	`drwx------`	`rw-------`
`$ umask 777`	`---rwxrwx`	Deny rwx to everyone (including the owner)	`d---------`	`----------`

Según la siguiente captura en valores octales sería: umask 050
El problema es que cuando quiero comprobarlo creando en /home un archivo cat me dice que no tengo permisos, así que tengo que usar sudo cat > ejemplo; pero entonces no me deja no sé si por que está bien el umask y entonces ya no me deja escribir o porque hay algún problema.

---

[2] KEIROZM.COM. “Default permissions for users, directories and files on linux: examples”[en línea]. *keirozm.com.* https://queirozf.com/entries/default-permissions-for-users-directories-and-files-on-linux-examples [consultado: 20 mayo 2021].

Each access level (read, write, execute) has an octal value:

Access level	Octal value
Read	4
Write	2
Execute	1

Each identity (user, group, others) has a position:

Identity	Position
User	First or left-most
Group	Middle
Others	Last or right-most

The absolute mode syntax states the desired permissions from left to right.

How do I grant the user (owner) read, write, and execute, the group read-only, and all others no access to `file2` by using absolute mode?

```
# chmod 740 file2
```

12. Give yourself and the members of your group complete permission on your files, but not allowing anyone (others) to do anything with them.

El comando sería umask 007