Linux Inodes

An Inode number is a uniquely existing number for all the files in Linux and all Unix type systems. When a file is created on a system, a file name and Inode number is assigned to it.

Generally, to access a file, a user uses the file name but internally file name is first mapped with respective Inode number stored in a table.

Note: Inode doesn't contain the file name. Reason for this is to maintain hard-links for the files.

Inode Contents

An Inode is a data structure containing metadata about the files. Following contents are stored in the Inode from a file:

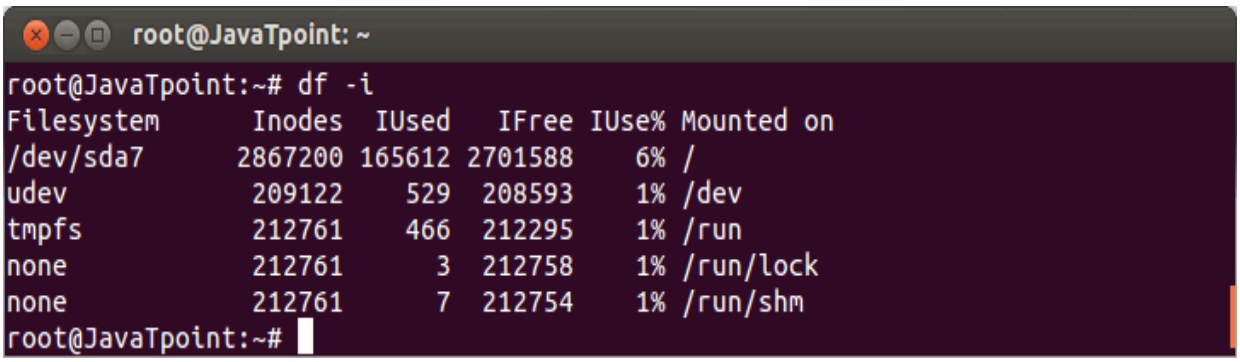
- User ID of file
- Group ID of file
- Device ID
- File size
- Date of creation
- Permission
- Owner of the file
- File protection flag
- Link counter to determine number of hard links

Inode Table

The Inode table contains all the Inodes and is created when file system is created. The

df -i

command can be used to check how many inodes are free and left unused in the filesystem.



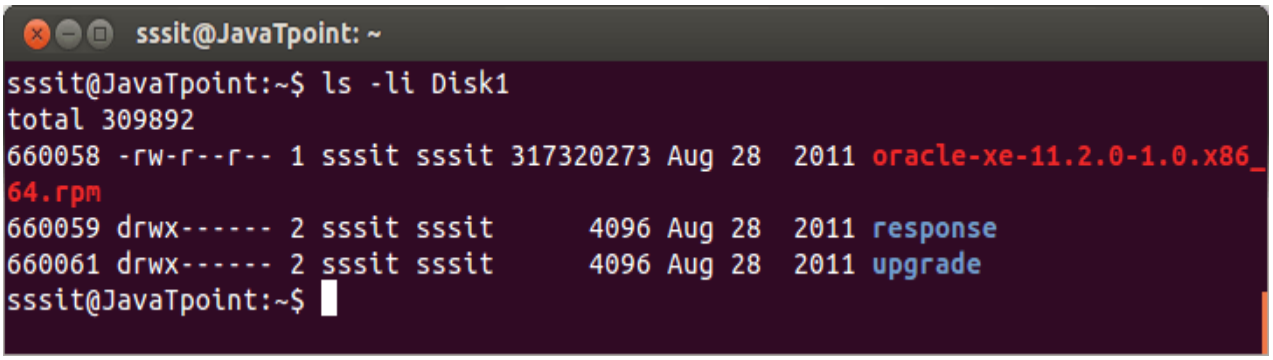
Look at the above snapshot, the command **"df -i"** shows the usage of several file systems.

Inode Number

Each Inode has a unique number and Inode number can be seen with the help of

ls -li

command.



Look at the above snapshot, Directory **Disk1** has the three files and each file has a different Inode number.

Note: The Inode doesn't contain file content, instead it has a pointer to that data.

Directories

- ### What is a Directory

A directory is a table which contains all its files Inode number and connect it to the file system.

- ### Dot (.) and DotDot (..)

The dot (.) is a mapping to itself and the dotdot (..) is a mapping to the parent directory.

Linux File Links

A Linux filesystem has many hard links and symbolic links. A link is a connectivity between the filename and the actual data byte in the disk space. More than one filename can **link** to the same data.

There are two types of links in Linux OS:

1. Soft Links
2. Hard Links

1) Soft Links (Symbolic Links)

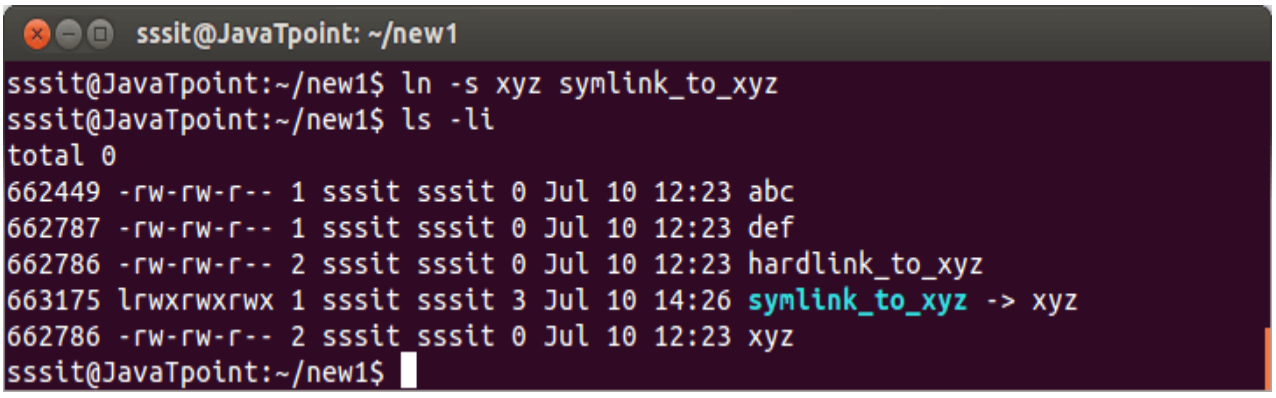
Soft links are very common. It represents a virtual or abstract location of the file. **It is just like the shortcuts created in Windows.** A soft link doesn't contain any information or content of the linked file, instead it has a pointer to the location of the linked file. In other words, a new file is created with new Inode, having a pointer to the Inode location of the original file.

When the source file is removed or moved, then soft links are not updated. We'll study in deep about both the links how to create it and remove it.

Symbolic links are also called **soft links.** Command **ln -s** is used to create soft link. It doesn't link to Inodes but create a name to mapping. It create its own Inode number.

Example:

```
ln -s xyz symlink_to_xyz
```



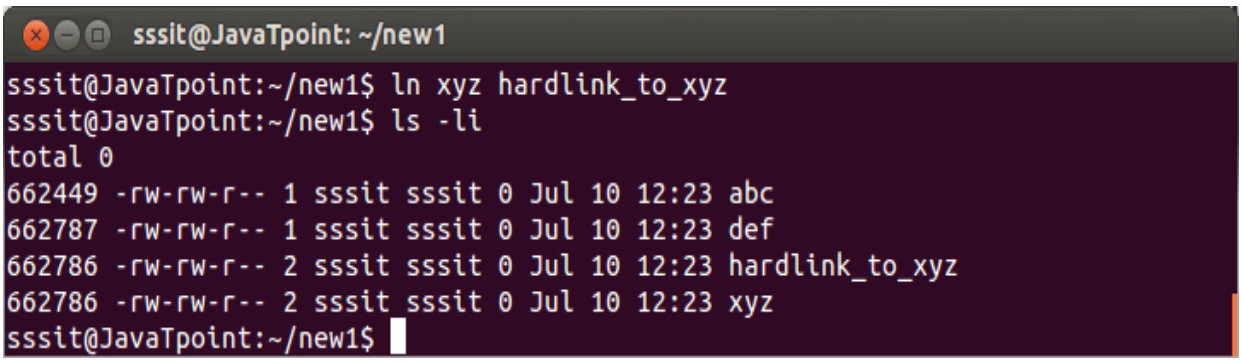
Look at the above snapshot, we have created a symbolic link for file **xyz** with command **"ln -s xyz symlink_to_xyz"**. Symbolic link Inode is different from the original file Inode number. Target permissions are applied on the symlink file. Hard links are limited to their own partition, but symbolic links can be linked anywhere.

2) Hard Links

They are the low-level links. It links more than one filename with the same Inode and it represents the physical location of a file. When hard link is created for a file, it directly points to the Inode of the original file in the disk space, which means no new Inode is created. Directories are not created using hard links and they can not cross filesystem boundaries. When the source file is removed or moved, then hard links are not affected.

Creating Hard Links

Hard links for any file can be created with command **ln**. One extra hard link file will be created in the respective directory.



Look at the above snapshot, we have created a hard link for the file **xyz** in the directory **new1.**

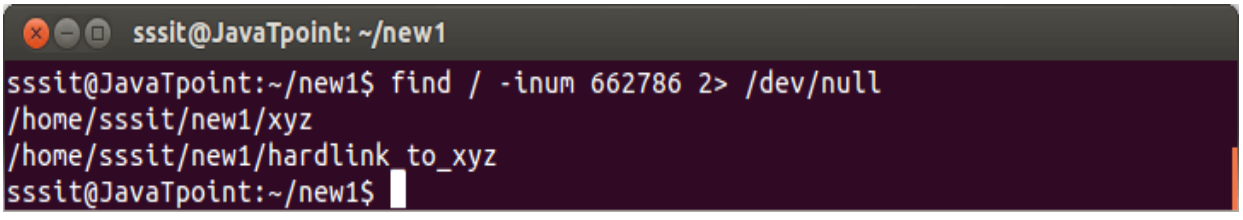
The original file and hard linked file both contain the same Inode number and hence, they have the same permissions and same owners. Content will also be the same for both the files. In short, both the files are equal now, but if original file will be removed then hard link file will not be affected.

Finding Hard Links

A hard link can be find with **find** command by specifying the Inode number. Inode number is always unique to its partition.

Example:

```
find / -inum 662786 2> /dev/null
```



Look at the above snapshot, we have found hard link files with command **"find / -inum 662786 2> /dev/null"** for the Inode number **662786.**