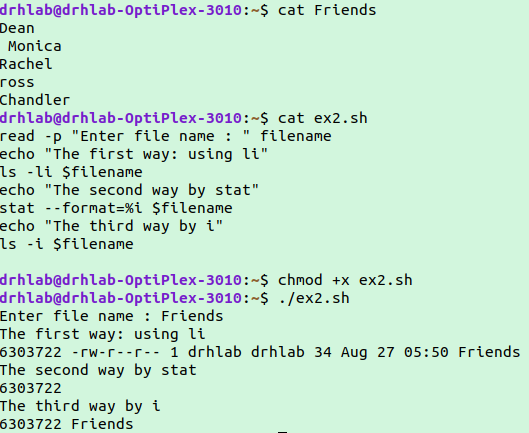
Inode in File System of Linux:

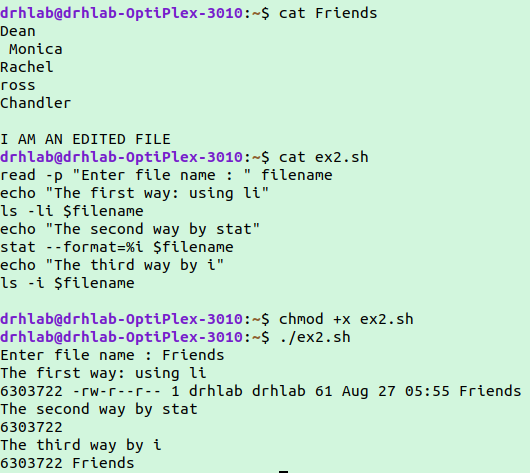
An inode is an entry in inode table, containing information (the metadata) about a regular file and directory. An inode is a data structure on a traditional Unix-style file system such as [ext3 or ext4](https://linoxide.com/how-tos/explained-in-detail-linux-ext2-ext3-and-ext4-filesystem/). Linux extended filesystems such as ext2 or ext3 maintain an array of these inodes: the inode table. This table contains list of all files in that filesystem. The individual inodes in inode table have a unique number (unique to that filesystem), the inode number. Diving deep into the inode, an inode stores:

* **File type:** regular file, directory, pipe etc.
* **Permissions to that file:** read, write, execute
* **Link count:**The number of hard link relative to an inode
* **User ID:** owner of file
* **Group ID:** group owner
* **Size of file:** or major/minor number in case of some special files
* **Time stamp**: access time, modification time and (inode) change time
* **Attributes:** immutable' for example
* **Access control list:** permissions for special users/groups
* **Link to location of file**
* **Other metadata** about the file

Here we will be using the command “chmod +x” to execute the file. I have explained “chmod” briefly in the file permission experiment. The use of this along with the filename makes the file executable. “./” runs this executable file. In the following screenshot, I have made a file in the VI editor whose inode number I want to know. In another while I have written 3 ways in which we can find the inode number of the file.



Even after editing the inode number of the file remains the same.



Comm Command and its implementation:

The **comm** command in Linux lets users compare two sorted files line by line.

The generalized syntax of the command is: ‘comm file1 file2’. This example shows how we have used comm command on two files.

*[](https://www.howtoforge.com/images/linux_comm_command/big/comm-output.png)*

So you can see that output consists of three columns. The first one contains lines unique to 'file1', second one contains lines unique to 'file2'and finally, column three contains lines common to both files.

If the files used in the command is unsorted then the common word is displayed along with an error of unsorted file. This can be solved by the command ‘sort filename’.

**Options**

|  |  |
| --- | --- |
| -1 | suppress column 1 (lines unique to FILE1) |
| -2 | suppress column 2 (lines unique to FILE2) |
| -3 | suppress column 3 (lines that appear in both files) |
| --check-order | check that the input is correctly sorted, even if all input lines are pairable |
| --nocheck-order | do not check that the input is correctly sorted |
| --outputdelimiter=*STR* | separate columns with [string](https://www.computerhope.com/jargon/s/string.htm)*STR* |
| --help | display a help message, and exit. |
| --version | output [version](https://www.computerhope.com/jargon/v/version.htm) information, and exit. |

I have created 2 files here and found out the common name in them using comm command.

