Contents

[**inode:** 2](#_Toc477347817)

[**No free inodes in file system:** 4](#_Toc477347818)

[1. Remove/move some zero sized files or files having very small size. 4](#_Toc477347819)

[2. Extend /resize file system 4](#_Toc477347820)

[3. Recreate file system with more inodes by specifying byte per inode (inode ratio): 4](#_Toc477347821)

[**Notes:** 4](#_Toc477347822)

# **inode:**

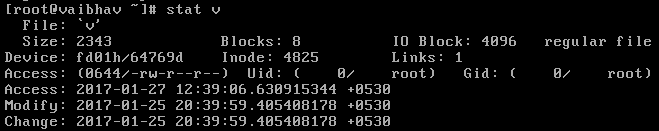
* Inode is a data structure which contains information about file **except the name of the file.**
* Inode contains information about file like
* file size
* file owner
* group to which file belongs to
* file permissions
* hard link count
* file location
* time stamp (last modified time[when actual data block was last changed], last access time, last changed time[when metadata was last changed])
* Inode number is a unique identification number associated with file. Inode number is a pointer which points to inode.
* Inode number is unique within partition. Files from different partition may have same inode number.
* When a file system is created, allocation is also done for the total number of inodes to be created in the file system. When new file is created, a new sequence number is created and is associated with that file.
* One inode number is assigned to file at the file creation time. If there are no free inodes, file can’t be created.
* There is a limit on maximum number of inodes which can be created so there is a limit on maximum number of files which can be created.
* Inode table contains inode numbers and corresponding file names.
* Inode stores metadata of files whereas Superblock stores metadata of file systems.
* ls -l(long list) command retrieves file information from inode only.
* To find out inode number of a file,

Command: ls -i <filename>



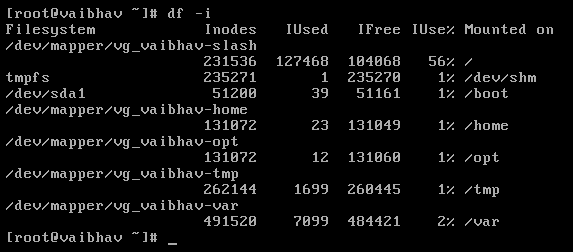
* To find out all information about inode,

Command: stat <filename>



* To find out total number of inodes, total inodes used, free inodes,

Command: df -i



* To find out file by its inode number,

Command: find -inum <inode number>

****

* **inode count** : maximum number of inodes that the file system can use
* **inode ratio**: ratio of inodes and bytes used at a file system creation
* Once file system is created, inode ratio is fixed and inode count can be changed by extending file system via LVM.
* The disk is full due to two reasons:

1. There is no physical memory space in disk
2. No free inodes

## **No free inodes in file system:**

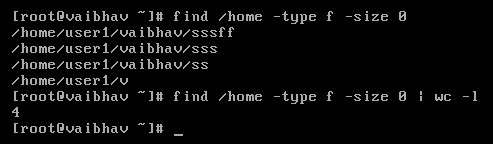
1. Remove/move some zero sized files or files having very small size.

When those files are removed/moved to other place, inodes corresponding to those file become free.

* To find zero sized file,

Command: find /home -type f -size 0

Command: find /home -type f -size 0 | wc -l: This will count of files having specified size



* To find small sized (1k) file,

Command: find /home -type f -size 1k

* To find large sized file,

Command: find /home -type d

### Extend /resize file system

To extend/ resize file system please refer document on file system mounting & LVM.

### Recreate file system with more inodes by specifying byte per inode (inode ratio):

* byte per inode =( (Block count)\*(block size) )/ (Inode count)
* Say we want to modify inode ration of /dev/mappers/vg\_vaibhav-opt file system and it is mounted on /opt.
* First of all, file system is already mounted so it needs to be unmounted first.

Command: umount /dev/mappers/vg\_vaibhav-opt

* Create the file system now,

Command: mkfs.ext4 -i <inode ratio> /dev/mappers/vg\_vaibhav-opt

* Now, mount the created file system.

Command: mount –t ext4 /dev/mappers/vg\_vaibhav-opt /opt

## **Notes:**

* inode count can be specified at the file system creation.
* Command: mkfs.<type of file system> -N <inode count> <file system>

Example: mkfs.ext4 -N 131072 /dev/mapper/vg\_vaibhav-opt