

Freeing Inode Usage

Credit [Michał Dąbrowski](#)

Introduction

In Unix-like systems, inodes are data structures that describe files and directories. The number of possible inodes is limited and set during partition creation. That means we can run out of them and be unable to create any new files, even if we have space on the device. In this tutorial, we'll learn how to prevent this situation and how to deal with it if it happens.

Why inodes Are Limited

Each inode contains crucial information about its file, like its attributes and disk block locations. This data is necessary for the system to use the file. In file systems from the family of extended file systems, the default for Linux-based systems, inodes are stored in a fixed-sized table. The size of this table is decided upon the creation of the partition and can't be changed later.

Other file systems (for example, APFS used on macOS) don't use fixed-sized tables but, instead, use other data structures like B-trees. Thus, the number of possible inodes is much more flexible. It's still limited by how big an index can be stored in a 64-bit integer (or 32-bit on older file systems), but that's a limit that's hard to hit.

Check and Free inodes

We can check the available inodes using the `df` command:

```
$ df -i
Filesystem      512-blocks      Used Available Capacity  iused      ifree %iused  Mounted on
/dev/disk1s5s1 489620264 46865488  34089872    58% 568975 2447532345    0% /
/dev/disk2s1   1953456384 727555584 1225900800   38% 2842014  4788675    37% /Volumes/T7
```

The "iused" column tells us the number of used inodes, "ifree" gives us the number of free inodes, and the "%iused" column tells us the percentage of used inodes.

Unfortunately, there is no way to free inodes other than deleting files we don't need. The problem is that we sometimes don't know where to look for files that drain the inode limit. One way to tackle this is to sort directories by the number of files in them. By doing so, we can quickly locate problematic directories.

We can achieve that by listing all the files, selecting only the first directory in the path, and then counting how many occurrences of each directory we have:

```
$ sudo find . -xdev -type f | cut -d "/" -f 2 | sort | uniq -c | sort -n
...
1585 Documents
6979 Downloads
7168 Pictures
113659 .nvm
669666 Library
980996 Projects
```

As we can see, we store the highest number of files in the "Projects" directory. At this moment, we can decide to investigate further or take some action like deleting files or moving that directory to some other drive.

Revision #1

Created 2024-01-28 13:39:08 UTC by Dino Edwards

Updated 2024-01-28 13:44:38 UTC by Dino Edwards