

Speaking UNIX: It's all about the inode

Learn how the UNIX file system manages files

Skill Level: Intermediate

[Adam T. Cormany \(acormany@yahoo.com\)](mailto:acormany@yahoo.com)

National Data Center Manager
Scientific Games Corporation

10 Jun 2008

Have you ever wondered what `Iused` and `%Iused` mean in UNIX® commands like `df` or what people are talking about when they say *inode*? UNIX and Linux® systems both use inodes, and IBM® AIX® is no different. Discover what an inode is and why inodes are important to UNIX, the structure of an inode, and commands for working with inodes.

An *inode* is a data structure in UNIX operating systems that contains important information pertaining to files within a file system. When a file system is created in UNIX, a set amount of inodes is created, as well. Usually, about 1 percent of the total file system disk space is allocated to the inode table.

Sometimes, people interchange the terms *inode* and *inumber*. The terms are similar and do correspond to each other, but they don't refer to the same things. *Inode* refers to the data structure; the *inumber* is actually the identification number of the inode—hence the term *inode number*, or *inumber*. The number is only one important item of information for a file. Some of the other attributes in an inode are discussed in the next section.

The inode table contains a listing of all inode numbers for the respective file system. When users search for or access a file, the UNIX system searches through the inode table for the correct inode number. When the inode number is found, the command in question can access the inode and make the appropriate changes if applicable.

Take, for example, editing a file with `vi`. When you type `vi <filename>`, the inode number is found in the inode table, allowing you to open the inode. Some attributes

are changed during the edit session of `vi`, and when you have finished and typed `:wq`, the inode is closed and released. This way, if two users were to try to edit the same file, the inode would already have been assigned to another user ID (UID) in the edit session, and the second editor would have to wait for the inode to be released.

The inode structure

The inode structure is relatively straightforward for seasoned UNIX developers or administrators, but there may still be some surprising information you don't already know about the insides of the inode. The following definitions provide just some of the important information contained in the inode that UNIX users employ constantly:

- Inode number
- Mode information to discern file type and also for the `stat` C function
- Number of links to the file
- UID of the owner
- Group ID (GID) of the owner
- Size of the file
- Actual number of blocks that the file uses
- Time last modified
- Time last accessed
- Time last changed

Basically, the inode contains all information about a file outside of the actual name of the file and the actual data content of the file. The full inode structure can be found in the header file `/usr/include/jfs/ino.h` in AIX or on the Web at

<http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/com.ibm.aix.files/doc/aixfiles/inoc>

The information listed above is important to files and is used heavily in UNIX. Without this information, a file would appear corrupt and unusable.

Directories and files may appear different on UNIX systems compared to other operating systems, but they aren't. In UNIX, directories are actually files that have a few additional settings in their inodes. A *directory* is basically a file containing other files. Also, the mode information has flags set to inform the system that the file is actually a directory.

Working with inodes

Knowing how to work with inodes in UNIX can save a lot of time and frustration. You can use the following commands to alleviate some of the headaches you may have when you don't know about inodes.

The df command

As mentioned earlier, when you create a file system in UNIX, about 1 percent of the total disk space is allocated to the inode table. Every time you create a file in the file system, an inode is allocated to the file. Typically, there is an adequate number of inodes associated with a file system, but running out of inodes is always a possibility. To monitor this, you can view the output of the `df`.

Using the `df` command, you can look at all mounted file systems or specific file systems. In this view, you can see the number of inodes used already in the respective file system as well as the percentage used overall in the file system, as [Listing 1](#) shows.

Listing 1. Using df to monitor inode use

```
# df -k|head -6
```

Filesystem	1024-blocks	Free	%Used	Iused	%Iused
Mounted on					
/dev/hd4	229376	138436	40%	4730	13%
/					
/dev/hd2	8028160	962692	89%	110034	33%
/usr					
/dev/hd9var	1835008	366400	81%	25829	24%
/var					
/dev/hd3	524288	523564	1%	98	1%
/tmp					
/dev/hd1	32768	32416	2%	5	1%
/home					

If for some reason a file system did reach 100 percent of its inodes used, you won't be able to create additional files, devices, directories, and so on in the file system. One solution is to add more space to the file system through the `smitty chfs` command, as shown in [Figure 1](#). Another solution is to create smaller inode extents. IBM AIX 5L now allows for inode extents smaller than the default size of 16KB on enhanced journal file systems. Please keep in mind, though, that if you use this option in AIX 5L, the file system will not be accessible from previous versions of AIX.

Figure 1. The result of the smitty chfs command

```

Change / Show Characteristics of an Enhanced Journaled File System

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[Entry Fields]
File system name      /usr
NEW mount point      [/usr]
SIZE of file system
  Unit Size          512bytes
  Number of units     [16056320]
Mount GROUP          [bootfs]
Mount AUTOMATICALLY at system restart?  yes
PERMISSIONS          read/write
Mount OPTIONS        {}
Start Disk Accounting?  no
Block Size (bytes)    1096
Inline Log?          no
Inline Log size (MBytes) [0]
Extended Attribute Format [v1]
ENABLE Quota Management?  no
ALLOW Small Inode Extents?  no

```

istat and stat

A quick way to examine an inode in AIX is by using the `istat` command. With this command, you can find the number of the specific file as well as other inode items like permissions; file type; UID; GID; number of links (not symbolic links); file size; and time stamps for last updated, last modified, and last accessed.

[Listing 2](#) shows inode information for the file `/usr/bin/ksh` in AIX.

Listing 2. Inode information for `/usr/bin/ksh`

```

# istat /usr/bin/ksh

Inode 18150 on device 10/8      File
Protection: r-xr-xr-x
Owner: 2(bin)                  Group: 2(bin)
Link count: 5                  Length 237804 bytes

Last updated:  Wed Oct 24 17:37:10 EDT 2007
Last modified: Wed Apr 18 23:58:06 EDT 2007
Last accessed:  Mon Apr 28 11:25:35 EDT 2008

```

In addition to showing the standard information from `istat`, you now know what the inumber is for `/usr/bin/ksh`. If you also find the logical volume in which the file resides, you can display even more information. One way to find this information is by looking at the mounted file system in which the file resides with the `df` command:

```

# df /usr/bin

Filesystem      512-blocks      Free %Used      Iused %Iused
Mounted on
/dev/hd2         16056320      1925384    89%      110034     33%
/usr

```

The file `/usr/bin/ksh` resides in the directory `/usr/bin`. Looking at the output of the `df` command, you can tell that the directory `/usr/bin` is contained in the `/usr` file system and that the `/usr` file system is inside the logical volume `/dev/hd2`. Now that you know both the inumber and the logical volume name, using `istat` with both items of

information as arguments, you can determine the hexadecimal addresses of the disk blocks that make up the file, as shown in [Listing 3](#).

Listing 3. Determining the hexadecimal addresses of the file blocks

```
# istat 18150 /dev/hd2

Inode 18150 on device 10/8      File
Protection: r-xr-xr-x
Owner: 2(bin)                  Group: 2(bin)
Link count: 5                  Length 237804 bytes

Last updated:   Wed Oct 24 17:37:10 EDT 2007
Last modified:  Wed Apr 18 23:58:06 EDT 2007
Last accessed:  Mon Apr 28 11:44:20 EDT 2008

Block pointers (hexadecimal):
11620          ef8c0
```

Linux has its own version of `istat`: `stat`. The Linux `stat` command shows similar information and also includes some switches not available in the AIX `istat` command:

```
# stat /bin/bash

File: `/bin/bash'
Size: 722684          Blocks: 1432          IO Block: 4096
regular file
Device: fd00h/64768d  Inode: 12799859      Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (  0/   root)   Gid: (
0/   root)
Access: 2008-04-06 19:13:50.000000000 -0400
Modify: 2006-07-12 03:11:53.000000000 -0400
Change: 2007-11-22 04:05:30.000000000 -0500
```

The ls command

At one time or another in your career, you've had to worry about removing or managing files with dashes or other special characters in the file name or file names that appear not to have a file name at all. Most likely, someone mistakenly named the respective file.

Because most commands in UNIX include switches, or options, that begin either with a hyphen (-) or a double hyphen (--), it can be difficult to manipulate these files with commonly used commands such as `rm`, `mv`, and `cp`. Thankfully, there are options in commands to show the inode number of the file in question. The `ls` command has such an option:

```
# ls
-      --      -p      fileA  fileB  fileC  fileD
fileE  fileF  fileG  fileH  fileI  fileJ  fileK  fileL
```

Using the `ls -i` command, you can view the inumber next to the file name, as shown in [Listing 4](#). Now that you know the inumber, you can easily manipulate the file.

Listing 4. Viewing the inumber of the file

```
# ls -i
38988      38991 -p      38984 fileC  38982 fileF  38977
fileI  38978 fileL
38989 -      38980 fileA  38986 fileD  38983 fileG  38987
fileJ
38990 --      38979 fileB  38976 fileE  38985 fileH  38981
fileK
```

The find command

Using the UNIX `find` command, you can finish what you started with the `ls` command. Now that you know the inumber for the respective files that you must manipulate, you can start!

To remove the file that looks like it has no name, simply use `find` with the `-inum` switch to locate the inumber and file. Then, when the file has been found, use `find` with the `-exec` switch to remove the file:

```
# find . -inum 38988 -exec rm {} \;
```

To rename the file, do the same again, but this time use `mv` rather than `rm`:

```
# find . -inum 38989 -exec mv {} fileM \;
```

To verify that you're getting the expected results, simply use the `ls -i` command again:

```
# ls -i
38990 --      38979 fileB  38976 fileE  38985 fileH  38981
fileK
38991 -p      38984 fileC  38982 fileF  38977 fileI  38978
fileL
38980 fileA  38986 fileD  38983 fileG  38987 fileJ  38989
fileM
```

The fsck command

Unfortunately, hardware doesn't last forever, and systems can fail over years of continued use. When this happens and the operating system shuts down abnormally because of a power failure or another issue, you may encounter files when you bring

the system back up that were open during the crash and now need assistance. During times like this, you may run into messages that inodes need to be repaired or that an error exists. If this happens, the `fsck` command can be a lifesaver! Rather than restoring the system or even rebuilding the operating system, you can use `fsck` to repair file systems or correct damaged inodes.

The following command attempts to repair the logical volume `/dev/hd1`:

```
# fsck -p /dev/hd1 -y
```

By using the `fsck` command, you can narrow the search for damaged inodes, as well. If you're searching for a specific inode, you can use the `-ii-NodeNumber` switch with `fsck`.

Conclusion

Files and directories would be nearly useless in UNIX without the helping hand of the inode. Hopefully, after reading this article, you understand inodes better, their importance to AIX, and also how to manage them. You may never look at `df` the same way again.

Resources

Learn

- [Speaking UNIX](#): Check out other parts in this series.
- [Inode on Wikipedia](#): Read more about inodes in Wikipedia's entry.
- [Wikipedia's AIX entry](#): Read Wikipedia's excellent entry on the AIX operating system for more information about its background and development.
- [The `istat` command](#): Learn more about the `istat` command from the Combined IBM Systems Information Center.
- [The `fsck` command](#): Learn more about the `fsck` command from the Combined IBM Systems Information Center.
- [The `df` command](#): Learn more about the `df` command from the Combined IBM Systems Information Center.
- [The `chfs` command](#): Learn more about the `chfs` command from the Combined IBM Systems Information Center.
- [The `ls` command](#): Learn more about the `ls` command from the Combined IBM Systems Information Center.
- [The `find` command](#): Learn more about the `find` command from the Combined IBM Systems Information Center.
- [The `smitty` command](#): Learn more about the `smitty` command from the Combined IBM Systems Information Center.
- [The `inode.h` file](#): Learn more about the `inode.h` file from the Combined IBM Systems Information Center.
- [The AIX and UNIX developerWorks zone](#) provides a wealth of information relating to all aspects of IBM® AIX® systems administration and expanding your UNIX skills.
- [New to AIX and UNIX?](#) Visit the New to AIX and UNIX page to learn more.
- [developerWorks technical events and webcasts](#): Stay current with developerWorks technical events and webcasts.
- [AIX Wiki](#): Visit this collaborative environment for technical information related to AIX.
- [Podcasts](#): Tune in and catch up with IBM technical experts.

Get products and technologies

- [IBM trial software](#): Build your next development project with software for

download directly from developerWorks.

Discuss

- Participate in the AIX and UNIX forums:
 - [AIX Forum](#)
 - [AIX forum for developers](#)
 - [Cluster Systems Management](#)
 - [IBM Support Assistant Forum](#)
 - [Performance Tools Forum](#)
 - [Virtualization Forum](#)
 - [More AIX and UNIX forums](#)

About the author

Adam T. Cormany

Adam Cormany is currently the manager of the National Data Center, but he has also been a UNIX systems engineer, a UNIX administrator, and operations manager for Scientific Games Corporation. Adam has worked extensively with AIX as well as in Solaris and Red Hat Linux administration for more than 10 years. He is an IBM eServer®-Certified Specialist in pSeries® AIX System Administration. In addition to administration, Adam has extensive knowledge of shell scripting in Bash, CSH, and KSH as well as programming in C, PHP, and Perl. You can reach Adam at acormany@yahoo.com.

Trademarks

IBM and AIX are registered trademarks of International Business Machines in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.