**Inodes and Disk Space**

Each file and directory needs an inode. Most filesystems are created with a fixed number of inodes, so if it runs out no more files can be stored even if there is free space.

The number of inodes, by default, is calculated as a ratio of bytes per inode, so a lower value gives you more inodes. If you would rather specify the exact number of inodes there is an option for that, which is seen shortly.

If a filesystem is to contain a lot of little files, such as many HTML or cache files, it should be formatted to contain many inodes so that more files can be stored. This is done using the smallest ratio available, effectively creating more inodes to handle the possible number of files. Practically, this would mean setting the bytes per inode to the same as the disk block size.

This topic is largely for the benefit of the exam. These days disks are so large that you get an incredible number of inodes. It’s rare to exhaust your inodes even at the default settings. The most common problem is that an application goes berserk and creates a lot of zero byte files without cleaning them up.

If you’re expecting to fill your disk with files smaller than 16K, the inode ratio may be an issue. Otherwise, you’re going to run out of disk space first.

The real danger of having a large block size and relatively few inodes (such as for a database server) is that you might run out of available inodes and still have disk space left over. A system can run completely out of inodes and still have available disk space, a situation that often requires extensive troubleshooting and work to remedy.

The inodes themselves take up 128 bytes of space each, or about 8,000 per megabyte of space. You must strike a balance between the number of files on disk and the size of the files. Too many inodes and you’re wasting space with inodes that never get used. Not enough and you’ll run out before the disk is full.

The **df -i** command shows inode utilization.

$ df -i  
Filesystem       Inodes   IUsed   IFree IUse% Mounted on  
/dev/mapper/VolGroup00-LogVol00  
                60686336 1500786 59185550  3% /

Here you can see that the root filesystem has about 60 million inodes and is using 3% of them.