8.4.5 File System Management Facts

The file system determines how a computer's files are organized on a hard drive.

This lesson covers the following topics:

- Linux file system types
- Formatting commands
- File system tips

Linux Files System Types

Linux supports many different file system types. The following table describes several common file systems.

Type	Description
ext2	The Second Extended File System (ext2) is one of the oldest Linux file systems still available. ▪ ext2 stores data in a standard directory and file hierarchy. ▪ The maximum file size supported is 2 TB. ▪ An ext2 volume can be up to 4 TB in size. ▪ File names can be up to 255 characters long. ▪ Linux users, groups, and permissions are supported. ▪ ex2 does not use journaling (which is used in most modern file systems). As a result, ext2 takes a long time to recover if the system shuts down abruptly.
ext3	The Third Extended File System (ext3) is an updated version of ext2 that supports journaling. Before committing a transaction to a storage device, the ext3 file system records the transaction to the journal and marks it as incomplete. After the disk transaction is complete, the file system marks the transaction as complete in the journal. By doing this, ext3 can keep track of the most recent file transactions and whether or not they were completed. This allows ext3 to recover much more quickly than ext2 in the event of an unclean system shutdown.
ReiserFS	The Reiser file system (ReiserFS) is an alternative to the ext3 file system. Like ext3, Reiser uses journaling to make crash recovery very fast. However, Reiser is a completely different file system from ext2 and ext3 and uses a dramatically different internal structure. ReiserFS supports a maximum file size of 8 TB and maximum volume size of 16 TB. In addition, the structure of Reiser allows it to perform much faster than ext2 or ext3.
ext4	ext4 is the fourth generation file system in the ext file system family. ext4 includes all of the features found with ext2 and ext3 with the addition of the following features: ▪ Support for file sizes up to 16 TB and disk sizes up to 1 exabyte (EB). ▪ Supports up to four billion files in the file system. ▪ Uses checksums to verify the integrity of the journal file itself. Checksums help improve the overall reliability of the system because the journal file is the most heavily used file of the disk.
swap	A swap file system is used as virtual memory (the portion of the hard disk used to temporarily store portions of main memory) by the operating system. A recommended practice is to make the swap file size between 1 and 1.5 times the amount of memory on the computer.
NTFS	Microsoft operating systems use NTFS (New Technology File System). Linux provides limited support for NTFS.
VFAT	VFAT is a FAT32 file system for Linux and does not support journaling. VFAT includes long name support. Support for VFAT must be compiled into the kernel for the system to recognize the VFAT format.
XFS	The XFS file system was developed for the Silicon Graphics IRIX operating system. An XFS file system is proficient at handling large files, offers smooth data transfers, and provides journaling. It also can reside on a regular disk partition or on a logical volume.
Btrfs	Btrfs is a Linux file system that uses a copy-on-write file system. Using copy-on-write technology, Btrfs provides several key features not found in earlier file systems, including storage pools and snapshots.

	▪ Instead of using traditional disk partitions, Btrfs allows you to create storage pools from the storage devices in your system. From the storage pool, you can then allocate space to specific storage volumes. Instead of mounting partitions, you mount storage volumes at mount points in the file system. ▪ The snapshot functionality provided by Btrfs protects data. It can be configured to take snapshots of your data at specified intervals and save it on separate media. If a file ever gets lost or corrupted, you can restore a previous version of the file from a snapshot.

Formatting Commands

A disk partition must be formatted using a file system. The following table describes the commands needed to format a partition.

Command	Function	Example

mkfs	Creates an ext family file system or a fat file system. The **mkfs** command uses the following options:
	▪ **-t *[file_system_type]*** determines the file system. File system types include:
	  ▪ **ext2** (identical to the **mkfs.ext2** command)
	  ▪ **ext3** (identical to **mkfs.ext3**)
	  ▪ **ext4** (identical to **mkfs.ext4**)
	  ▪ **vfat** (identical to **mkfs.vfat**)
	  ▪ **reiserfs** (identical to **mkreiserfs**)
	▪ **-b** specifies the block size. Supported values are 1024, 2048, or 4096.
	▪ **-i** determines how many inodes are on the partition and uses the same values as **-b**.
	▪ **-j** appends a journal to an ext2 file system.
	Without the **-b** and **-i** options, **mkfs** calculates the optimal values for you automatically.
		▪ **mkfs -t ext2 /dev/sda4** creates an ext2 file system on the fourth partition on the first hard disk drive.
		▪ **mkfs -t ext3 /dev/sda1** creates an ext3 file system on the first partition on the first hard disk drive.
		▪ **mkfs -t ext3 /dev/sdc2** creates an ext3 file system on the second partition on the third hard disk drive.
		▪ **mkfs -t ext4 /dev/sdb1** creates an ext4 file system on the first partition on the second hard disk drive.

mkreiserfs	Creates a ReiserFS.	▪ **mkreiserfs /dev/sda2** formats the second partition on the first hard disk with the Reiser file system.
mkswap	Creates a swap partition. A swap partition is the location on the hard drive where an operating system writes memory information when it runs out of RAM. ▪ The **swapon** command must be run to activate the swap partition. ▪ The **swapoff** command is used to deactivate swap partitions. Both **swapon** and **swapoff** use the **-a** option to enable or disable all swap partitions listed in **/etc/fstab**.	▪ **mkswap /dev/sda2** formats the second hard drive as the swap partition. ▪ **swapon /dev/sda2** activates the second hard drive as the swap partition. ▪ **swapon -a** activates all swap partitions. ▪ **swapoff /dev/sda2** deactivates the second hard drive as the swap partition. ▪ **swapoff -a** deactivates all swap partitions.
mke2fs	Create an ext2, ext3, or ext4 file system. Command options include the following: ▪ **-b** specifies the block size of the file system in bytes (valid sizes are 1024, 2048, and 4096 bytes per block). ▪ **-j** creates the file system with an ext3 journal. ▪ **-L** sets the volume label for the file system. ▪ **-n** displays what **mke2fs** would do if it created a file system, but does not actually create the file system. ▪ **-t** specifies the file system type (such as ext2, ext3, or ext4) to be created.	▪ **mke2fs /dev/sda2** creates an ext2 file system on the second partition on the first hard disk drive. ▪ **mke2fs -j /dev/sda1** creates an ext3 file system on the first partition on the first hard disk drive. ▪ **mke2fs -t ext4 /dev/sdc3** creates an ext4 file system on the third partition on the third hard disk drive.

Files System Tips

Keep the following in mind when working with file systems:

- You cannot format an extended partition. However, you can create logical partitions inside an extended partition and format them.
- File systems use an inode (information node) table to store information about files. An inode specifies where a file's data physically exists on a disk. Inodes also contain additional information, including:

    - File size
    - Modification, access, and creation times
    - Permissions
    - Ownership
- Each file system has a superblock, which contains information about the file system, such as:
    - File system type (such as ext2 and ext3)
    - Size (for example, 10GB or 360GB)
    - Status
- Linux maintains multiple redundant copies of the superblock in every file system.
