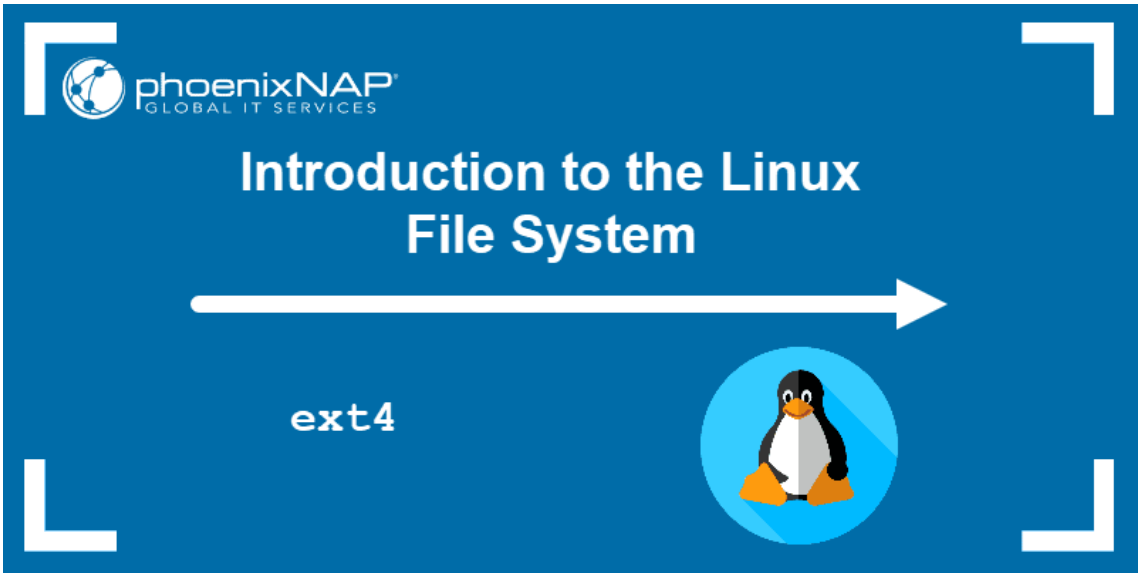


Introduction

A file system is a set of processes that controls how, where and when data is stored and retrieved from a storage device. An efficient file system is essential for everyday system processes.

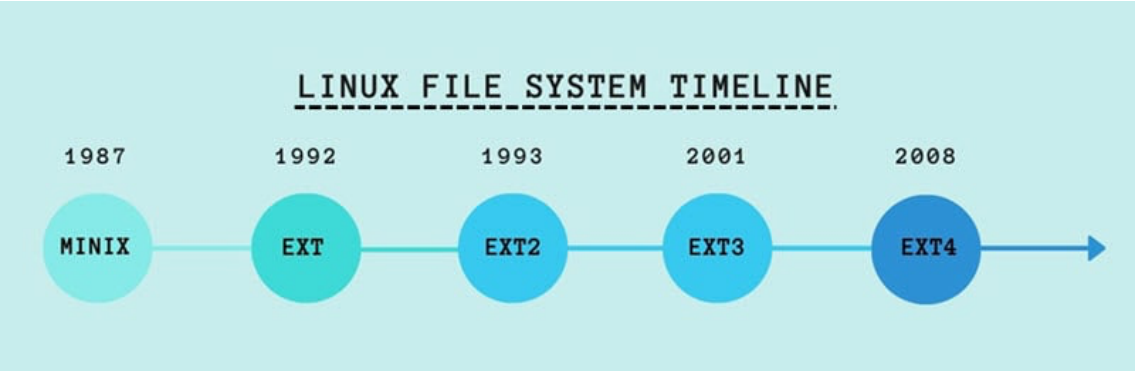
The Linux kernel supports various file systems, but the most commonly used is the ext4 file system.

In this article, you will learn more about the development of Linux file systems and the main features of the ext4 system.

Note: In Linux, everything is stored as a file (e.g., directories, printers, partitions, kernel data, etc.). That makes it all the more important to learn how the Linux file system works.

Linux File System Evolution

Let’s take a closer look at the evolution of the Linux file system:



MINIX File System

The Minix file system supported the Minix operating system. It was first introduced in 1987 by Andrew S. Tanenbaum.

The Minix operating system and its file system were mostly used for educational coding purposes. The performance of the file system was not up to standard at the time. Filename lengths were restricted to fourteen characters, and partitions were limited to 64MB. At the time, hard drives supported partitions up to 140MB.

By 1992 Minix was mostly out of use due to lack of performance and development of the **ext file system**.

ext File System

The ext file system stands for “Extended File System”. It was the first file system designed to support the Linux kernel.

Virtual File System (VFS) was used for the ext file system. Its primary purpose was to allow the Linux kernel to access the ext file system. The ext file system restricted filename lengths to 255 characters and supported partitions up to 2GB.

While it managed to solve issues that the Minix file system had, it had one major flaw – timestamping. Unlike today where each Linux file has three timestamps (access timestamp, modified timestamp, and changed timestamp), the ext file system allowed only one timestamp per file.

In January 1993, the ext2 file system was introduced. In time, all users switched from ext to ext2.

ext2 File System

Remi Card designed the **ext2 file system** and released it in January 1993, less than a year after introducing the ext file system.

The ext2 file system enabled the retention of the internal structure while the file system functionalities extended. Data from files were kept in data blocks of the same length. The ext2 file system supported the maximum file size of 2TiB. Filename lengths were not limited in characters, but in bytes – 255 bytes. It did not support journaling.

While this file system was largely used, it still had two major issues:

- **File corruption** – This phenomenon would occur if data were written to the disk at the time of a power loss or system crash.
- **Performance loss** – Disk fragmentation happens when a single file is broken into pieces and spread over several locations on the disk. As a result, files take longer to read and write, which leads to performance degradation.

The ext2 system was mostly used until the early 2000s when the ext3 file system was introduced. It is occasionally used today for USB devices because it does not support the journaling system.

ext3 File System

Stephen Tweedie designed the **ext3 file system** (Third Extended File System). It launched in November 2001 with Linux kernel 2.4.15. It is still in use today.

The ext3 file system is an improved version of ext2 file system. It supports a maximum file size of 2TiB and restricts maximum filename length to 255 bytes, like the ext2 file system. The improvement is reflected in journaling.

The **journaling** system keeps a “journal” of all changes in the data structure that are yet to be commited. In case of power loss or system crash, logs stored via the journaling system return data in a manner of seconds, reducing the risk of corruption or data loss. The system writes the data in the correct areas of the file system when the log is updated.

The Linux kernel supports three levels of journaling:

- **Journal** – It consists of writing metadata and file contents in a journal before changes are made to the main file system. This saves data in case of a power loss or system crash. The disadvantage of this level of journaling is that the performance of the system declines.

- **Ordered** – This journaling level writes the metadata to the journal, while file contents are automatically stored in the main file system. The process is performed in a specific order. First, the metadata is written in the journal. Then, the file contents are written to the main file system. Eventually, the metadata connects to the main file system. Therefore, the main file system is not corrupted in the event of a system crash. Only files that are in the process of being written during a crash can be corrupted.
- **Writeback** – This level of journaling only writes metadata to the journal. File contents are written to the main file system only after the journal is updated. Due to the lack of synchronization of metadata and file content, the file system will likely be corrupted if the system crashes.

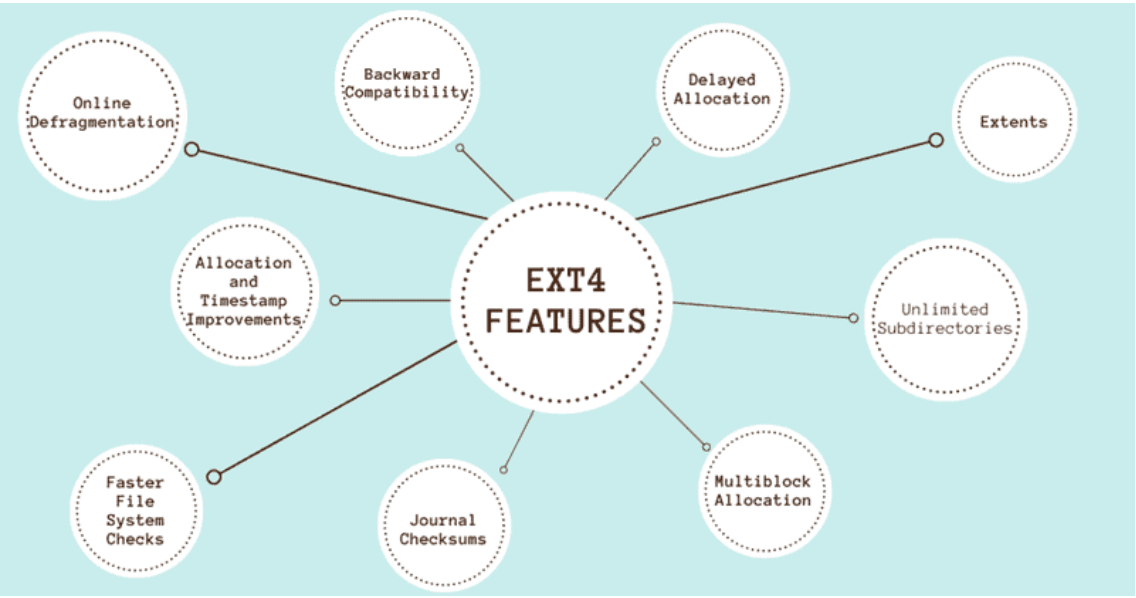
ext4 File System

The ext4 file system is the default file system of the current Linux kernel. It was introduced in October 2008 with Linux kernel 2.6.28.

The ext4 file system supports the maximum file size of 16TiB and restricts maximum filename lengths to 255 bytes.

Features of the ext4 File System

Let's look at the main features of the ext4 file system.



Backward Compatibility

The ext4 file system supports backward compatibility with ext3 and ext2 file systems. An additional feature is the automatic mounting of the ext3 file system in ext3 mode by using an ext4 driver.

Allocation Improvements

The ext4 file system allocates storage blocks more efficiently prior to writing them to the disk. That improves read and write performance.

Timestamp Improvements

The ext4 file system adds another 408 years to the timestamp and supports dates up to May 10, 2446. Timestamps are also measured faster, in nanoseconds.

Extents

Outdated versions of the ext file system map all the blocks that correlate with each file. The process does not work when it comes to large files that require a high number of blocks. Extents have resolved the problem in the ext4 file system.

Extents reduce the amount of metadata required to map each file's blocks. The system saves the address of the first and last block corresponding to the large file.

Multiblock Allocation Improvements

A block allocator searches for free blocks that can be used to write data to the disk. The ext4 file system uses **multiple allocations** that allow the allocation of multiple blocks per call. This reduces disk fragmentation.

Delayed Allocation

The **delayed allocation** feature allocates blocks only when the file is written to the disk. With this feature, cache memory is not filled with unnecessary data, and the performance of the system increases.

Unlimited Number of Subdirectories

Linux kernel version 2.6.23 supports an unlimited number of subdirectories. The ext4 file system introduced the **HTree data structure** to avoid drops in performance. The HTree data structure represents a specialized version of the B-tree.

Journal Checksums

The ext4 file system uses the **checksum** option. This option was introduced to reduce the risk of file corruption.

The journaling system is the most used part of the disk. When hardware failure occurs, the blocks become unusable and file corruption occurs.
The checksum option constantly checks to see if a block is damaged. This process also improves performance because it shortens the journaling time.

Faster File System Checks

In an ext4 file system, undistributed groups of blocks and inode tables are marked. The time required to run the `fsck` command is significantly shortened because marked groups are skipped. It improves overall performance.

Online Defragmentation

Disk fragmentation leads to performance degradation, which was a significant issue with the ext2 and ext3 file systems. The ext4 file system supports the **e4defrag** tool that lets users defragment individual files or the complete file system.

Limitations of ext4 File System

Although the ext4 file system is considered as the best file system for Linux distributions, there are a few limitations that should be considered in the further development of the system:

- **Corrupted data recovery** – The ext4 file system cannot detect or recover corrupted data already written on the disk.
- **Maximum volume size** – The maximum volume size is set to 1 EiB. However, the file system cannot address more than 100 TiB of data without a significant loss of performance and increased disk fragmentation.

Alternative Linux File Systems

There are several alternatives to the ext4 file system. The Linux kernel supports all the alternatives listed below.

Note: To avoid issues, use alternative file systems under separate directories.

XFS

XFS is a 64-bit file system that was first introduced in 1994 and built into the Linux kernel since 2001. It is the default file system for RedHat Linux.

XFS supports a maximum file size of 8 EiB and restricts filename length to 255 bytes. It supports journaling, and like ext4, it saves the changes in a journal before changes are committed to the main file system. This reduces the possibility of file corruption.

Data is structured in **B+ trees**, which provides efficient space allocation and therefore increased performance.

This system's main disadvantage is reflected in the difficult resizing process of an existing XFS file system.

OpenZFS

OpenZFS is a platform that combines file systems with volume managers. It was first introduced in 2013.

OpenZFS supports a maximum file size of 16 EiB and limits the maximum filename length to 255 characters. Some of this system's features are protection against data corruption, encryption, support for high storage capacities, copy-on-write, and RAID-Z.

OpenZFS' main disadvantage is the legal incompatibility between CDDL (OpenZFS) and GPL (Linux kernel) licenses. That is resolved by compiling and loading the ZFS code to the Linux kernel.

BtrFS

Oracle designed **BtrFS** (stands for "B-tree file system") and released it in 2009 with Linux kernel 2.6.29.

BtrFS supports a maximum file size of 16 EiB and limits the maximum filename length to 255 characters. Some of the BtrFS' features are online defragmentation, online block device addition and removal, RAID support, compression configurable per file or volume, file cloning, checksums, and the ability to handle swap files and swap partitions.

Conclusion

The Linux file system evolved for decades to gain its complexity and functionality. Each new functionality resolved an issue present in the outdated versions of the system.

After reading this article, you should have a better understanding of the Linux file system and how it functions.