**Министерство науки и высшего образования Российской Федерации** федеральное государственное автономное образовательное учреждение высшего образования

# «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

**Дисциплина:**

«Операционные системы»

# Лабораторная работа №5

“Тестирование ФС”

Выполнил:

Хоанг Вьет Зунг – Студент группы N32491

Нгуенг Хонг Ньят - Студент группы N32451

Преподаватель: Савков С.В.



Санкт-Петербург 2022

# Задание:

**Простое вариант:**

Выбрать 3 (или больше) файловых систем, выбрать методику проверки и найти лучшую из них.

# Усложненный вариант

Экзотические фс или Экзотические методики проверки

For this laboratory work, I will test these following file systems: ext3, ext4, ntfs and btrfs (для усложенного вариант)

1. Journal is a log that records changes to the file system before they are actually committed to the actual file system. Journaling is a technique used by file systems to ensure data consistency in the event of power failures, system crashes, or other unexpected events that could cause data loss or corruption.

When a file system uses journaling, it writes changes to the file system to a log called a journal, which is typically located at the beginning of the file system. The journal records each change to the file system, including metadata changes (such as file creation and deletion, changes to file permissions, and changes to file attributes) and changes to file data. There are a few journaling modes available for some of the file systems in Linux:

* Journal: In this mode, all file system data and metadata changes are written to the journal before they are committed to the file system. This provides the highest level of data consistency and protection but can result in lower performance due to the additional overhead of writing to the journal.
* Ordered: In this mode, only metadata changes are written to the journal immediately. File data changes are written to the file system first and then to the journal in a specific order to ensure consistency.
* Ordered: In this mode, only metadata changes are written to the journal immediately. File data changes are written to the file system first and then to the journal in a specific order to ensure consistency.

1. **ext3**: The ext3 file system is an improvement over the earlier ext2 file system. It provides better reliability and faster recovery in case of system failures. It supports large file sizes and can handle file systems up to 16 terabytes in size. The ext3 file system uses an inode structure to keep track of files and directories. Inodes are data structures that contain information about files and directories, such as ownership, permissions, timestamps, and file size. The file system also uses a block allocation method to allocate disk space to files.
2. **ext4:** Ext4 is the successor to the ext3 file system and offers several improvements in terms of performance, scalability, and reliability. One of the key features of ext4 is its support for larger file systems and files. Ext4 file systems can support file sizes of up to 16 terabytes and file systems up to 1 exabyte in size. This allows ext4 to support larger data sets and workloads than its predecessor.
3. **Ntfs:** NTFS is a robust and feature-rich file system that offers several advantages over the older FAT file system, including support for larger file sizes, more efficient use of disk space, and more advanced security features. In Linux, NTFS support is provided by the NTFS-3G driver, which allows Linux systems to read and write to NTFS file systems. NTFS-3G supports many of the advanced features of NTFS, including support for large files, file compression, encryption, and advanced permissions and security settings.
4. Btrfs (B-tree file system) is a modern file system developed to address some of the limitations of older file systems such as ext4 and XFS. Btrfs was designed with scalability, reliability, and data integrity in mind, and it offers several advanced features that are not found in other file systems. One of the key features of Btrfs is its support for snapshots and copy-on-write functionality, which allows for efficient and fast backups and restores of data. Btrfs also supports subvolumes, which are independent file system structures that can be used to organize and manage data more efficiently.

\*Method dd: The dd command reads from or writes to a file and can be used to test the throughput of a file system by reading and writing large files. It can also be used to test the I/O performance of a storage device by reading or writing data to specific disk blocks.

\*Method Fio:

- Fio (Flexible IO Tester) is a benchmarking and performance testing tool that can be used to test the performance of file systems by simulating different types of I/O workloads, such as sequential or random reads and writes, and measuring the performance metrics such as I/O throughput, latency, and IOPS (input/output operations per second). Fio also supports a wide range of parameters and options that can be used to customize the test scenarios, including the type and size of the data being tested, the number of threads and processes used, and the type of I/O operations being performed.

-  There are some of the key parameters of Fio:

+ iodepth: controls the number of outstanding I/O requests that can be submitted to the storage device in parallel.

+ numjobs: controls the number of threads that are used to perform I/O operations.

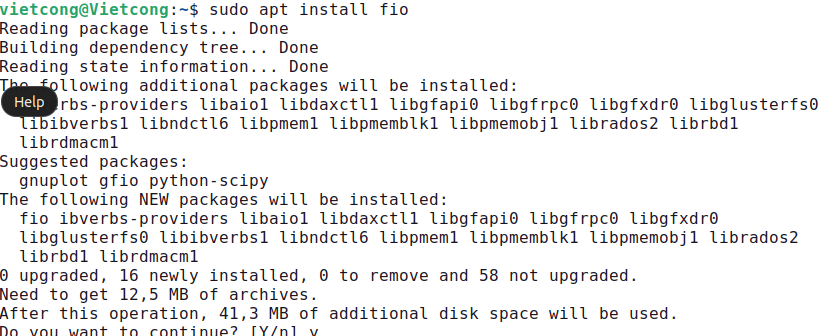
+ runtime: The runtime parameter controls the length of time that the test runs.

+ bs: controls the block size of the I/O operations

First we need to install btrfs and fio

Text, letter

Description automatically generated



Then we create a bash file to test the file systems:

#bin/bash!

for T in "ext3" "ext4" "ntfs" "btrfs";do

#dd

sudo dd if=/dev/zero of=$T.txt bs=1M count=64

sudo mkfs -t $T -F $T.bin

#fio

sudo fio --filename=$T.bin --ioengine=libaio --direct=1 --time\_base --group\_reporting --runtime=5 --size=64MiB --eta-newline=1 --bs=4k --iodepth=256 --numjobs=4 --rw=randrw --name=$T-randrwriops >> ./$T-iops.txt

sudo fio --filename=$T.bin --ioengine=libaio --direct=1 --time\_base --group\_reporting --runtime=5 --size=64MiB --eta-newline=1 --bs=64k --iodepth=54 --numjobs=4 --rw=randrw --name=$T-randrwrthroughput >> ./$T-throughput.txt

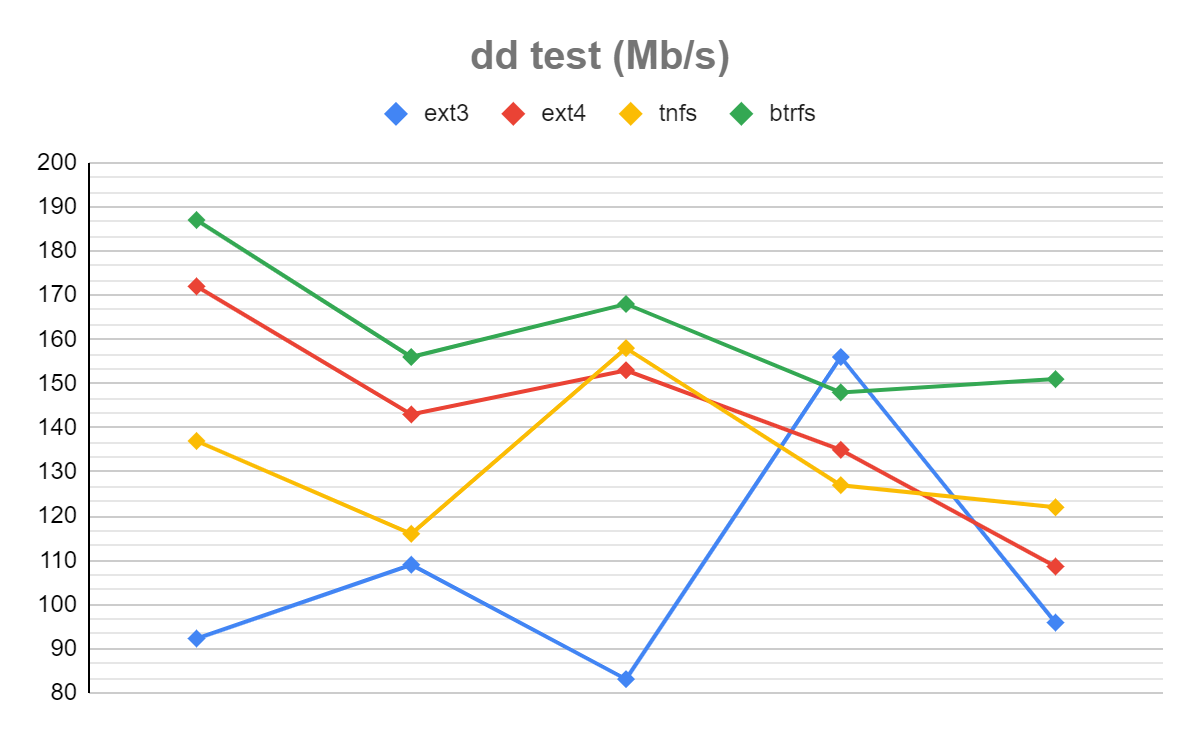
echo "-------------"

done

\*\*Result:

Table

Description automatically generated



**\*dd test file systems (Mb/s) :** btrfs > ext4 > tnfs > ext3

**Table

Description automatically generated**

**Chart, bar chart

Description automatically generated**

**Fio Test (MiB):** btrfs > ext4 > ext3 > tnfs

 Conclusion: in general, the performance of the file systems ext4 and btrfs seems to be better than ext3 and ntfs. This is because they support advanced features such as delayed allocation and copy-on-write, which can improve the efficiency of I/O operations. Ext4 is a modern file system that is designed to improve the performance and scalability of the earlier ext3 file system. It uses several advanced features, such as extents and delayed allocation, to improve file system performance. Btrfs is also a modern copy-on-write file system that supports advanced features such as snapshots, compression, and checksumming. On the other hand, NTFS is a file system used by Windows operating systems, and it is not natively supported by Linux. While it is possible to use NTFS on Linux using third-party tools, the performance may not be as good as native file systems such as ext4 and btrfs.