

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Факультет безопасности информационных технологий

Дисциплина:
«Операционные системы»


ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5
«Тестирование файловых систем»

Выполнили:

Бардышев Артём Антонович,
студент группы N3246

(подпись)

Суханкулиев Мухаммет,
студент группы N3246



(подпись)

Проверил:

Савков Сергей Витальевич,
инженер

(отметка о выполнении)

(подпись)

Санкт-Петербург
2024 г.

СОДЕРЖАНИЕ

Введение	3
1 Тестирование файловых систем	5
1.1 Создание, монтирование и тестирование файловой системы	5
1.2 Тестирование разных файловых систем двумя способами	6
1.3 Анализ результатов	6
Заключение	8
Список использованных источников	9

ВВЕДЕНИЕ

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

Усложнение:

Экзотические файловые системы или экзотические методики проверки.

Основные определения:

Файловые системы для Linux

1. **EXT (Extended File System)** — первая файловая система, разработанная специально для Linux, с улучшенной производительностью по сравнению с предыдущими решениями.
2. **EXT2** — расширенная версия EXT.
3. **EXT3** — журналируемая версия EXT2.
4. **EXT4** — улучшенная эффективность, надежность, поддержка больших объемов хранения.
5. **BTRFS (B-Tree File System)** — разработана Oracle; использует структуру СУБД, обеспечивает отказоустойчивость и поддержку сжатия.
6. **F2FS (Flash-Friendly File System)** — от Samsung, оптимизирована для флэш-памяти. F2FS разбивает носитель на части, которые снова делятся, и так далее. Эти миниатюрные зоны используются вместо повторного использования одних и тех же размеченных участков.
7. **ReiserFS** — журналируемая ФС от Namesys разработана под руководством Ганса Райзера, позволяет изменять размер на лету, однако эта ФС может показать нестабильные результаты и потерять данные, например, при отключении энергии.
8. **XFS** — от SGI, известна производительностью и масштабируемостью, особенно для серверов.
9. **JFS** — 64-битная ФС от IBM с высокой производительностью и надежностью.

Файловые системы для других ОС

1. **HFS+** — файловая система от Apple, улучшенная версия HFS для Mac OS. Во время разработки эта система называлась Sequoia.
2. **NTFS** — от Microsoft для Windows NT, с журналированием изменений метаданных.
3. **FAT32** — версия FAT для DOS и Windows; основная файловая система для флешек, а также внешних HDD и SSD, однако в настоящее время наблюдается тенденция

отказа от FAT32 в пользу более продвинутых файловых систем, таких как NTFS, exFAT, Ext2/Ext3.

4. **exFAT** — расширенная версия FAT32, оптимизирована для больших файлов на съемных накопителях.

Специализированные файловые системы

1. **OpenZFS** — ответвление от ZFS, разработчик — компания Sun для ОС Solaris. В 2016 году Ubuntu включила ее поддержку по умолчанию. Главные плюсы: защита от повреждения данных, поддержка больших файлов и автоматическое восстановление.
2. **tmpfs** — временная ФС, записывает файлы в оперативную память.
3. **procfs** — содержит информацию о системных процессах и ядре.
4. **sysfs** — позволяет изменять настройки ядра ОС.

Виртуальные файловые системы

1. **EncFS** — шифрует файлы и сохраняет их в зашифрованном виде (основана на FUSE).
2. **Aufs (Another Union File System)** — слоистая ФС, объединяет несколько каталогов в один; часто применяется в Live CD и Docker.
3. **NFS (Network File System)** — удаленное монтирование файловых систем.
4. **ZFS** — разработана для Solaris, поддерживает управление пулами хранения, снимками, клонированием и шифрованием.

Инструменты тестирования файловых систем

1. **Iozone** — измеряет скорость файловой системы при помощи различных операций с файлами.
2. **Bonnie++** — фокусируется на открытии/закрытии файлов, что влияет на производительность файловой системы. То есть производительность диска она особо не тестирует.
3. **fio** — выполняет операции над файлами или устройствами, то есть мы можем исключить файловую систему из рассмотрения. Включает режимы randwrite, randread, randrw.

FUSE (Filesystem in USErspace) — интерфейс для Unix-систем, позволяющий создавать пользовательские ФС без редактирования кода ядра. По сути, модуль FUSE предоставляет мост к фактическим интерфейсам ядра.

1 ТЕСТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

1.1 Создание, монтирование и тестирование файловой системы

Установка инструментов для создания и управления разными файловыми системами

```
apt install -y e2fsprogs btrfs-progs xfsprogs f2fs-tools reiserfsprogs jfsutils  
dosfstools ntfs-3g
```

```
dd if=/dev/zero of=ext4.img bs=1M count=512  
mkfs -t ext4 -F ext4.img  
mount ext4.img /mnt/test  
iozone -a /mnt/test/  
bonnie++ -d /mnt/test -s 128 -r 64 -u root  
umount ext4.img
```

Ключевые параметры **iozone**

- a – Автоматический режим: запускает все тесты с разными размерами блоков и файла.
- s <size> – Размер тестируемого файла (например, -s 512M).
- r <record size> – Размер блока (например, -r 4k).
- i <type> – Тип теста:
 - 0 – запись
 - 1 – повторная запись
 - 2 – случайное чтение/запись
 - 3 – обратное чтение
 - 4 – обратная запись
- f <filename> – Задаёт файл для теста.
- t <threads> – Количество потоков.
- T – Включает мониторинг CPU.
- c – Выводит информацию о кэшировании.

Ключевые параметры **bonnie++**

- d <directory> – Директория для теста.
- s <size> – Размер тестового файла (обычно больше объема ОЗУ).
- r <RAM size> – Размер ОЗУ для настройки тестов.
- u <user> – Пользователь для теста.
- n <num_files>:<min_size>:<max_size>:<increment> – Тестирование множества файлов с заданным диапазоном и шагом размеров.

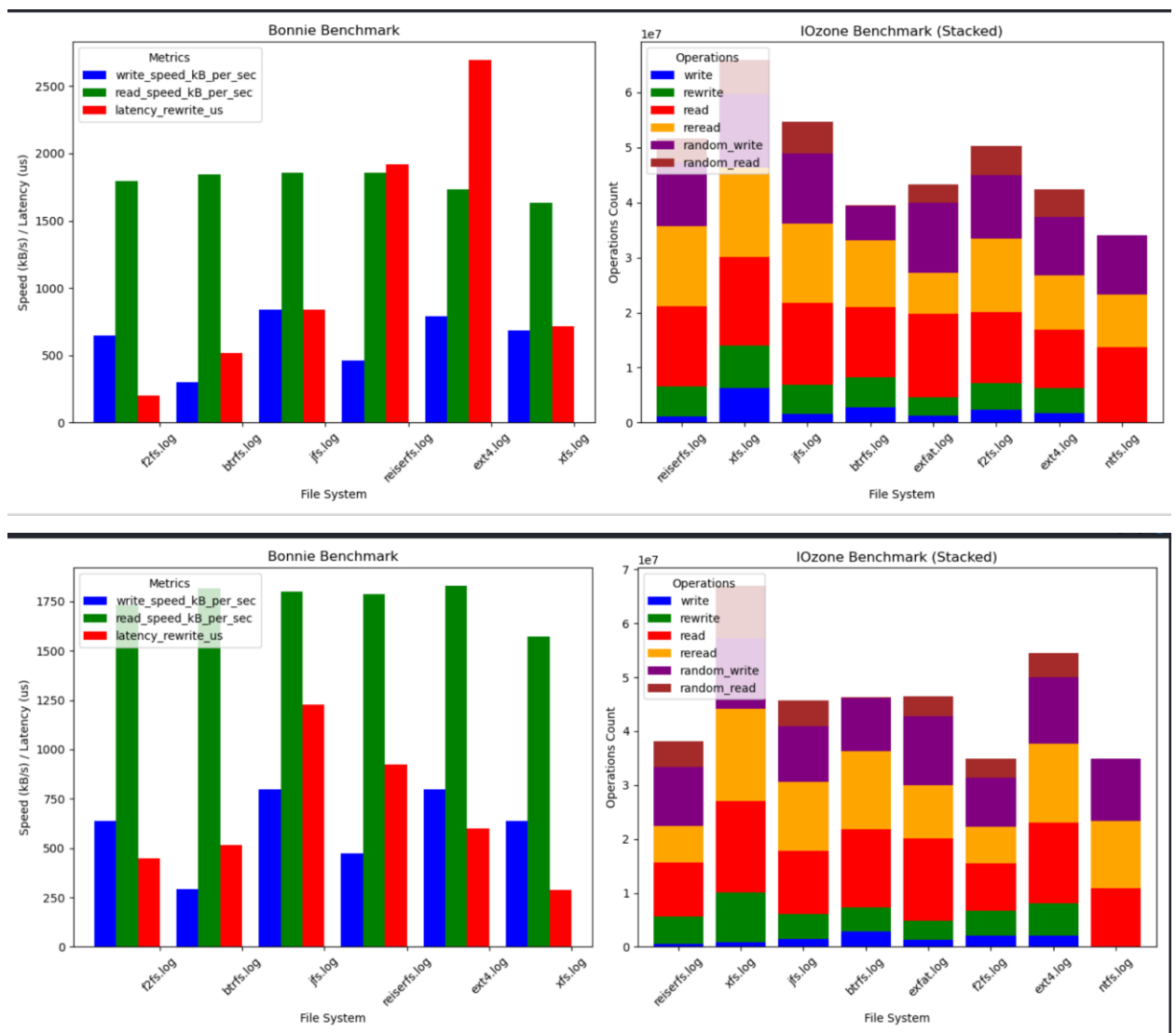
- m <label> – Метка теста для отчетов.
- x – Расширенный режим с несколькими циклами.
- q – Тихий режим с минимальным выводом.

1.2 Тестирование разных файловых систем двумя способами

Мы написали скрипт для выполнения лабораторной работы и вывода всей информации в .log файлы.

Затем с Python мы преобразуем логи в csv формат и выводим нужные диаграммы.

1.3 Анализ результатов



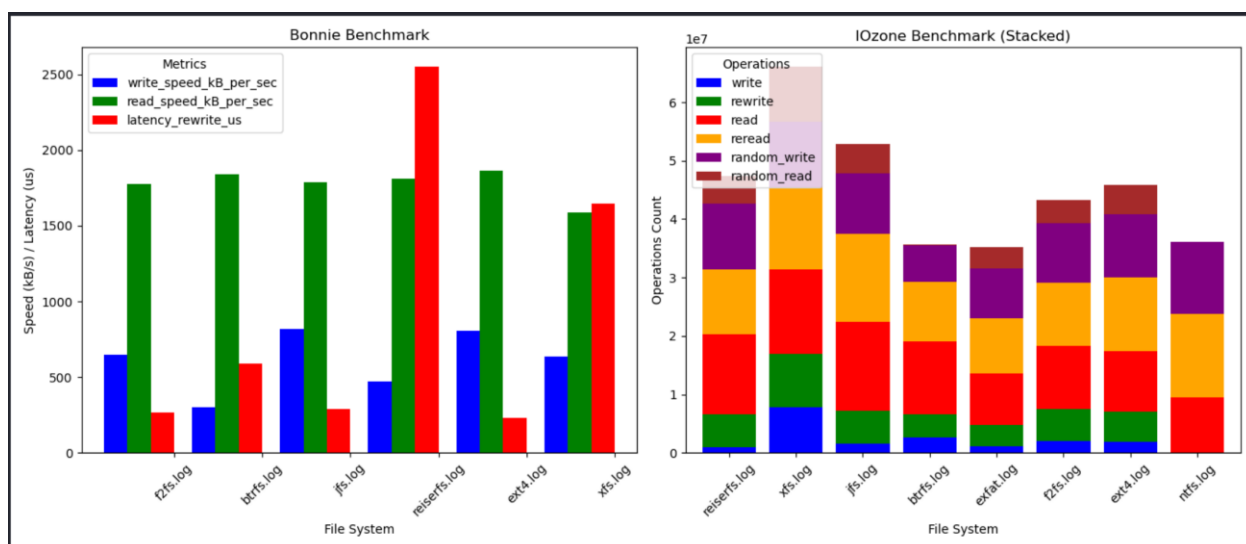


Рисунок 1 – Результаты тестирования при помощи bonnie++ и iohome (с накоплением)

По графикам видим, что при тестировании iohome **лучшие** показатели имеет **XFS**, затем идут **JFS**, **EXT4** и **BTRFS**. Худшие результаты у **NTFS**.

При тестировании bonnie++ (ntfs и exfat не были протестированы) видно, что наименьшую задержку показывает **F2FS**, а лучшие чтение/запись у **JFS** и **EXT4**, худшие результаты у **BTRFS** и **ReiserFS**.

ЗАКЛЮЧЕНИЕ

В ходе лабораторной работы тестирование показало, что для нашей виртуальной машины **XFS** и **JFS** являются самыми производительными файловыми системами по большинству показателей. Они демонстрируют отличные результаты как при записи, так и при чтении на различных типах накопителей. В то же время, файловые системы **NTFS**, **BTRFS** и **ReiserFS** имеют ограничения по скорости и задержке, что делает их менее подходящими для задач, требующих высокой производительности.

Однако, **EXT4** и **F2FS** могут быть оптимальным выбором для пользователей, которым важен баланс между производительностью и низкой задержкой, особенно в условиях использования SSD и флэш-накопителей. В специфических сценариях, несмотря на ограничения, более старые файловые системы, такие как **FAT32** и **exFAT**, также могут быть полезными.

Результаты тестирования подтверждают, что новые и хорошо оптимизированные файловые системы, такие как **F2FS** и **XFS**, показывают высокую производительность на различных типах накопителей. Тем не менее, выбор файловой системы всегда должен учитывать конкретные условия эксплуатации, включая тип накопителя, размер файлов и требования к скорости записи/чтения. Важно выбрать наиболее подходящую файловую систему в зависимости от потребностей системы и рабочих условий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. [Тестирование двенадцати файловых систем в Linux](#)
2. [Testing Disk Performance on Linux | Baeldung on Linux](#)
3. [Как правильно мерять производительность диска / Хабр](#)
4. [iozone\(1\): Filesystem Benchmark - Linux man page](#)
5. [Тестирование производительности жесткого диска с помощью Bonnie++](#)
6. [Структура и типы файловых систем в Linux - Академия Selectel](#)
7. [Файловые системы в Linux: их структура и типы](#)
8. [FUSE: как написать свою файловую систему / Хабр](#)
9. [What is a FUSE filesystem? | Jan 2023 | Goncalo Amaral | Medium](#)
10. [GitHub - libfuse/libfuse: The reference implementation of the Linux FUSE \(Filesystem in Userspace\) interface](#)
11. [GitHub - fusepy/fusepy: Simple ctypes bindings for FUSE](#)