

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Факультет безопасности информационных технологий**

**Дисциплина: «Операционные системы»  
ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5**

**Выполнил:**

Студент группы N3249

Чан Нгок Хуан



**Проверил:**

Савков Сергей Витальевич

Санкт-Петербург

2022г.

## Лаб 5. Тестирование ФС.

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

Усложненный вариант

Экзотические фс или Экзотические методики проверки

Задания на паре

1. С помощью dd создать диск

```
dd if=/dev/zero of=123.bin bs=1M count=64
```

2. С помощью mkfs создать файловую систему mkfs -t ntfs -F 123.bin

3. Смонтировать фс в папку

```
mount 123.bin mymount/
```

```
echo dfg > mymount/123.txt
```

```
umount 123.bin
```

4. losetup /dev/loop0 disk.img

5. fdisk disk.img

создать 2 диска

6. kpartx -a 123.bin

создать фс на обоих дисках, смонтировать, создать файлы, размонтировать,

```
kpartx -d 123.bin
```

7. lvm тома

создать диск из двух файлов

```
dd if=/dev/zero of=./d01 count=1 bs=1M # записать файл данными
```

```
losetup -f --show ./d01 # смонтировать файл как устройство
```

```
pvccreate /dev/... # сделать доступным
```

```
Pvmove # освободить от данных диск
```

Группа дисков

```
vgscan # какие группы есть
```

```
vgcreate -s 32M [name] disks # создать группу дисков
```

```
vgextend, vgreduce # добавить и удалить из группы
```

Логический диск

```
lvcreate -n first -L 2G vg # создать логический том
```

```
lvresize -L 40G vg1/lv2 # изменить размер тома
```

```
Lvremove # удаление тома
```

## I.Файловые системы

**Файловая система** – это инструмент, позволяющий операционной системе и программам обращаться к нужным файлам и работать с ними. При этом программы оперируют только названием файла, его размером и датой создания. Все остальные функции по поиску необходимого файла в хранилище и работе с ним берет на себя файловая система накопителя.

Основные функции файловой системы:

- Фрагментация файлов и их распределение на носителе.
- Поиск файла при запросе программ.
- Участие в создании, чтении и удалении файлов.
- Работа с атрибутами файлов: изменение названия, размера, времени последнего изменения, доступ к файлу и многое другое.
- Каталогизация и организация файлов.
- Защита файлов от несанкционированного доступа и сбоев системы.
- Определение права доступа к файлам.
- Восстановление информации в случае сбоев.

Таким образом, файловая система устанавливает правила эксплуатации и организацию данных на накопителе, и тем самым экономит ресурсы операционной системы и рабочих программ. К тому же наличие файловой системы позволяет использовать накопитель на разных компьютерах без каких-либо предварительных настроек и оптимизации.

**Журналируемая файловая система** — файловая система, в которой осуществляется ведение журнала, хранящего список изменений и, в той или иной степени, помогающего сохранить целостность файловой системы при сбоях.

Я буду тестировать 4 файловые системы: **ext3, ext4, NTFS, XFS** (Экзотические фс )

### 1. Ext3

Файловая система Ext3 – являет собой доработанную версию Ext2. Добавилось журналирование – технология, которая позволяет значительно снизить риск потери информации. Суть заключается в следующем — файловая система отмечает все планируемые изменения в структуре данных. В случае потери питания система мгновенно считывает информацию с «журнала» и файлы не будут утеряны.

Ext3 поддерживает три уровня журналирования:

Journal – файловая система записывает и метаданные и пользовательские данные в журнал, тем самым практически гарантируя успешное восстановление файловой системы в случае потери питания. В то же время этот метод наиболее сильно снижает производительность файловой системы.

Writeback – сначала записываются метаданные в журнал и только после этого информация записывается в файловую систему. Синхронизация метаданных и данных в файловой системе отсутствует. Этот уровень журналирования является самым быстрым, из-за отсутствия

необходимости проверки. Однако в этом заключается главный недостаток – если что-то случится с системой — вы потеряете данные.

Ordered (упорядоченное журналирование) – процесс записи выполняется в определенном порядке: сначала записываются метаданные в журнал, а после этого пользовательские данные записываются в файловую систему. После этого, метаданные подключаются к файлам на диске. В случае отключения питания «исчезнут» только те файлы, которые записывались на диск в этот момент. Структура файловой системы не повреждается.

## 2. Ext4

Ext4 — журналируемая файловая система, используемая преимущественно в операционных системах с ядром Linux, созданная на базе ext3 в 2006 году.

Ext4 – как и Ext3 имеет обратную совместимость с предыдущими версиями ФС. Собственно говоря, вы можете смонтировать Ext2 или Ext3 как Ext4 -и при определённых условиях добиться большей производительности. Вы так же можете смонтировать Ext4 как Ext3 без каких-либо побочных эффектов.

Является первой ФС из “семейства” Ext, использующая механизм “extent file system”, который позволяет добиться меньшей фрагментации файлов и увеличить общую производительность файловой системы. Кроме того – в Ext4 реализован механизм отложенной записи (delayed allocation – delalloc), который так же уменьшает фрагментацию диска и снижает нагрузку на CPU. С другой стороны – хотя механизм отложенной записи и используется во многих ФС – но в силу сложности своей реализации он повышает вероятность утери данных.

К плюсам Ext4 можно отнести следующие:

- Журналирование – файловая система ведет журнал изменений, благодаря чему в случае сбоев вы без проблем можете восстановить структуру файловой системы;
- Поддержка шифрования – теперь пользователи могут шифровать свои данные без падения производительности;
- Высокая стабильность – файловая система Ext4 может сама контролировать свое состояние. Сделано это при помощи контрольных сумм журналов. Кроме того, с момента релиза Ext4 прошло более 10 лет, и за это время она зарекомендовала себя как очень надежная файловая система;
- Поддержка по умолчанию во многих дистрибутивах – что означает, что весь инструментарий для работы с Ext4 у вас будет из коробки. Кроме того, она используется в картах памяти смартфонов на Android, а значит вам не нужно будет устанавливать дополнительный софт для работы с данными со смартфона;
- Активная разработка – на сегодняшний день разработчики активно работают над улучшением Ext4. Вы будете постоянно получать новые возможности для работы с данными;
- Низкий уровень фрагментации – вы получаете очень быструю файловую систему. Часто именно отсутствие фрагментации является решающим фактором, когда речь заходит какую файловую систему использовать на флешках и съемных носителях;

- Большое количество лимитов – во время создания файловой системы создается некоторое количество inodes, которое необходимо для записи файлов. В некоторых файловых системах количество inodes может оказаться слишком малым, и пользователь не может записать новый файл даже при наличии свободного места на диске. Обычно это актуально при работе с большим количеством маленьких файлов. К счастью Ext4 лишена этой проблемы и количества inodes хватает как для обычных пользователей, так и для серверных систем

К сожалению, минусов у файловой системы, Ext4 тоже хватает. Среди наиболее значимых можно выделить:

- Отсутствие поддержки функций файловых систем следующего поколения – в Ext4 отсутствует поддержка управления томами, дедупликация данных и т.д;
- Отсутствие проверки контрольных сумм для данных – из-за этого невозможно обнаружить повреждения данных из-за аппаратных сбоев оборудования;
- Плохая масштабируемость – несмотря на то, что максимальный размер раздела составляет 1 Экзабайт, в реальности при создании разделов размером больше 100 Терабайт очень сильно падает производительность;
- Отсутствие поддержки прозрачного сжатия и прозрачного шифрования – эти технологии пока что находятся на экспериментальной стадии;
- Inodes занимают до 10% объема раздела – если во время создания файловой системы будет выделено меньшее количество inodes – существует вероятность что они закончатся и тогда пользователь не сможет записать новые файлы даже при наличии свободного места на диске;

### **3. NTFS**

- NTFS - стандартная файловая система для семейства операционных систем Windows.
- Каталог на NTFS представляет собой специфический файл, хранящий ссылки на другие файлы и каталоги, создавая иерархическое строение данных на диске.
- NTFS поддерживает хранение метаданных. Информация о файлах хранится в главной файловой таблице — Master File Table (MFT). Для повышения надёжности файловой системы в NTFS используется система журналирования USN.

### **4. XFS**

XFS - это высокомасштабируемая файловая система, разработанная Silicon Graphics и впервые развернутая в операционной системе IRIX на базе Unix в 1994 году. Это файловая система с журналированием которая отслеживает изменения в журнале перед фиксацией изменений в основной файловой системе. Преимущество заключается в гарантированной целостности файловой системы и ускоренном восстановлении в случае сбоев питания или сбоев системы.

Одной из примечательных особенностей XFS является гарантированная скорость ввода-вывода. Это позволяет приложениям зарезервировать пропускную способность. Файловая система рассчитывает доступную производительность и корректирует свою работу в соответствии с существующими резервированиями.

XFS имеет репутацию системы, работающей в средах, требующих высокой производительности и масштабируемости, и поэтому регулярно оценивается как одна из самых производительных файловых систем в больших системах с корпоративными рабочими нагрузками.

Сегодня XFS поддерживается большинством дистрибутивов Linux и теперь стала файловой системой по умолчанию в Red Hat Enterprise Linux, Oracle Linux, CentOS и многих других дистрибутивах.

#### Сравнения нескольких ограничений

Файловая система	Родная ОС	Максимальная длина имён файлов	Максимальный размер файла	Максимальный размер тома
Ext3	Linux	255 байт	16 ГиБ — 2 ТиБ	2 ТиБ — 32 ТиБ
Ext4	Linux	255 байт	16 ГиБ до 16 ТиБ	1ЭиБ
NTFS	Windows NT	255 символов	16ЭиБ	16 ЭиБ
XFS	IRIX	255 байт	9ЭиБ	9ЭиБ

## II. Проверка

lozone is a filesystem benchmark tool. The benchmark generates and measures a variety of file operations. lozone has been ported to many machines and runs under many operating systems. This document will cover the many different types of operations that are tested as well as coverage of all of the command line options.

lozone is useful for determining a broad filesystem analysis of a vendor's computer platform. The benchmark tests file I/O performance for the following operations.

Read, write, re-read, re-write, read backwards, read strided, fread, fwrite, random read/write, pread/pwrite variants.

```
-i # Test to run (0=write/rewrite, 1=read/re-read, 2=random-read/write
    3=Read-backwards, 4=Re-write-record, 5=stride-read, 6=fwrite/re-fwrite
    7=fread/Re-fread, 8=random_mix, 9=pwrite/Re-pwrite, 10=pread/Re-pread
    11=pwritev/Re-pwritev, 12=preadv/Re-preadv)
```

Я буду тестировать с **Read, write, re-read, re-write** (i = 0 и i = 1)

Write: This test measures the performance of writing a new file. When a new file is written not only does the data need to be stored but also the overhead information for keeping track of where the data is located on the storage media. This overhead is called the “metadata” It consists of the directory information, the space allocation and any other data associated with a file that is not part of the data contained in the file. It is normal for the initial write performance to be lower than the performance of rewriting a file due to this overhead information.

Re-write: This test measures the performance of writing a file that already exists. When a file is written that already exists the work required is less as the metadata already exists. It is normal for the rewrite performance to be higher than the performance of writing a new file.

Read: This test measures the performance of reading an existing file.

Re-Read: This test measures the performance of reading a file that was recently read. It is normal for the performance to be higher as the operating system generally maintains a cache of the data for file that were recently read. This cache can be used to satisfy reads and improves the performance.

#### Установка iozone:

```
(huan@kali)-[~]
└─$ sudo apt-get update
[sudo] password for huan:
Get:1 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Get:2 http://packages.microsoft.com/repos/code stable/main arm64 Packages [82.4 kB]
Get:4 http://packages.microsoft.com/repos/code stable/main amd64 Packages [81.5 kB]
Get:5 http://packages.microsoft.com/repos/code stable/main armhf Packages [82.3 kB]
Hit:3 http://mirror-1.truenetwork.ru/kali kali-rolling InRelease
Fetched 257 kB in 1s (347 kB/s)
Reading package lists... Done

(huan@kali)-[~]
└─$ sudo apt-get install iozone3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following package was automatically installed and is no longer required:
  libqt5opengl5
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be installed:
  iozone3
0 upgraded, 1 newly installed, 0 to remove and 2264 not upgraded.
Need to get 422 kB of archives.
After this operation, 724 kB of additional disk space will be used.
Get:1 http://mirror-1.truenetwork.ru/kali kali-rolling/non-free amd64 iozone3 amd64 489-1 [422 kB]
Fetched 422 kB in 1s (424 kB/s)
Selecting previously unselected package iozone3.
(Reading database ... 412808 files and directories currently installed.)
Preparing to unpack .../iozone3_489-1_amd64.deb ...
Unpacking iozone3 (489-1) ...
Setting up iozone3 (489-1) ...
Processing triggers for kali-menu (2021.2.0) ...
Processing triggers for man-db (2.9.4-2) ...
```

- Тестирование с ФС ext3:

С помощью dd создать диск :

```
(huan@kali)-[~]
└─$ dd if=/dev/zero of=ext3.bin bs=8M count=64
64+0 records in
64+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 2.67262 s, 201 MB/s
```

С помощью mkfs создать файловую систему

```
(huan@kali)-[~]  
$ mkfs -t ext3 -F ext3.bin  
mke2fs 1.46.2 (28-Feb-2021)  
Discarding device blocks: done  
Creating filesystem with 131072 4k blocks and 32768 inodes  
Filesystem UUID: 09b0bae2-926f-4624-b62d-beee54a4be48  
Superblock backups stored on blocks:  
    32768, 98304  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (4096 blocks): done  
Writing superblocks and filesystem accounting information: done
```

Создать папку и смонтировать фс в папку

```
(huan@kali)-[~]  
$ sudo mkdir /home/huan/OS/LAB5/ext3  
  
(huan@kali)-[~]  
$ sudo mount ext3.bin /home/huan/OS/LAB5/ext3
```

Тестировать с помощью iotzone

```
(huan@kali)-[~]  
$ iotzone -A /home/huan/OS/LAB5/ext3 -b ext3.xls | tee ext3.txt  
Iotzone: Performance Test of File I/O  
Version $Revision: 3.489 $  
Compiled for 64 bit mode.  
Build: linux-AMD64
```

- Тестирование с ФС ext4:

```
(huan@kali)-[~]  
$ dd if=/dev/zero of=ext4.bin bs=8M count=64  
64+0 records in  
64+0 records out  
536870912 bytes (537 MB, 512 MiB) copied, 5.56035 s, 96.6 MB/s  
  
(huan@kali)-[~]  
$ mkfs -t ext4 -F ext4.bin  
mke2fs 1.46.2 (28-Feb-2021)  
Discarding device blocks: done  
Creating filesystem with 131072 4k blocks and 32768 inodes  
Filesystem UUID: 6f602867-3479-4b96-a942-855776fdfe4e  
Superblock backups stored on blocks:  
    32768, 98304  
  
Allocating group tables: done  
Writing inode tables: done  
Creating journal (4096 blocks): done  
Writing superblocks and filesystem accounting information: done
```



```
(huan@kali)-[~]  
$ sudo mkdir /home/huan/OS/LAB5/ext4  
  
(huan@kali)-[~]  
$ sudo mount ext4.bin /home/huan/OS/LAB5/ext4
```

```
(huan@kali)-[~]  
$ iozone -A /home/huan/OS/LAB5/ext4 -b ext4.xls | tee ext4.txt  
Iozone: Performance Test of File I/O  
Version $Revision: 3.489 $  
Compiled for 64 bit mode.  
Build: linux-AMD64  
  
Contributors: William Norcott, Don Capps, Isom Crawford, Kirby Collins  
Al Slater, Scott Rhine, Mike Wisner, Ken Goss  
Steve Landherr, Brad Smith, Mark Kelly, Dr. Alain CYR,  
Randy Dunlap, Mark Montague, Dan Million, Gavin Brebner,  
Jean-Marc Zucconi, Jeff Blomberg, Benny Halevy, Dave Boone,  
Erik Habbinga, Kris Strecker, Walter Wong, Joshua Root,  
Fabrice Bacchella, Zhenghua Xue, Qin Li, Darren Sawyer,  
Vangel Bojaxhi, Ben England, Vikentsi Lapa,  
Alexey Skidanov, Sudhir Kumar.  
  
Run began: Sat Apr 30 15:11:43 2022  
  
Auto Mode 2. This option is obsolete. Use -az -i0 -i1  
Command line used: iozone -A -b ext4.xls /home/huan/OS/LAB5/ext4  
Output is in kBytes/sec  
Time Resolution = 0.000001 seconds.  
Processor cache size set to 1024 kBytes.  
Processor cache line size set to 32 bytes.  
File stride size set to 17 * record size.
```

- Тестирование с ФС NTFS:

```
(huan@kali)-[~]  
$ dd if=/dev/zero of=ntfs.bin bs=8M count=64  
64+0 records in  
64+0 records out  
536870912 bytes (537 MB, 512 MiB) copied, 5.18246 s, 104 MB/s
```

```
(huan@kali)-[~]  
$ mkfs -t ntfs -F ntfs.bin
```

```
(huan@kali)-[~]  
$ sudo mkdir /home/huan/OS/LAB5/ntfs
```

```
(huan@kali)-[~]  
$ sudo mount ntfs.bin /home/huan/OS/LAB5/ntfs
```

```
(huan@kali)-[~]
$ iozone -A /home/huan/OS/LAB5/ntfs -b ntfs.xls | tee ntfs.txt
Iozone: Performance Test of File I/O
Version $Revision: 3.489 $
Compiled for 64 bit mode.
Build: linux-AMD64
```

- Тестирование с ФС XFS:

```
(huan@kali)-[~]
$ sudo apt-get install xfsprogs
```

```
(huan@kali)-[~]
$ dd if=/dev/zero of=xfs.bin bs=8M count=64
64+0 records in
64+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 1.71683 s, 313 MB/s
```

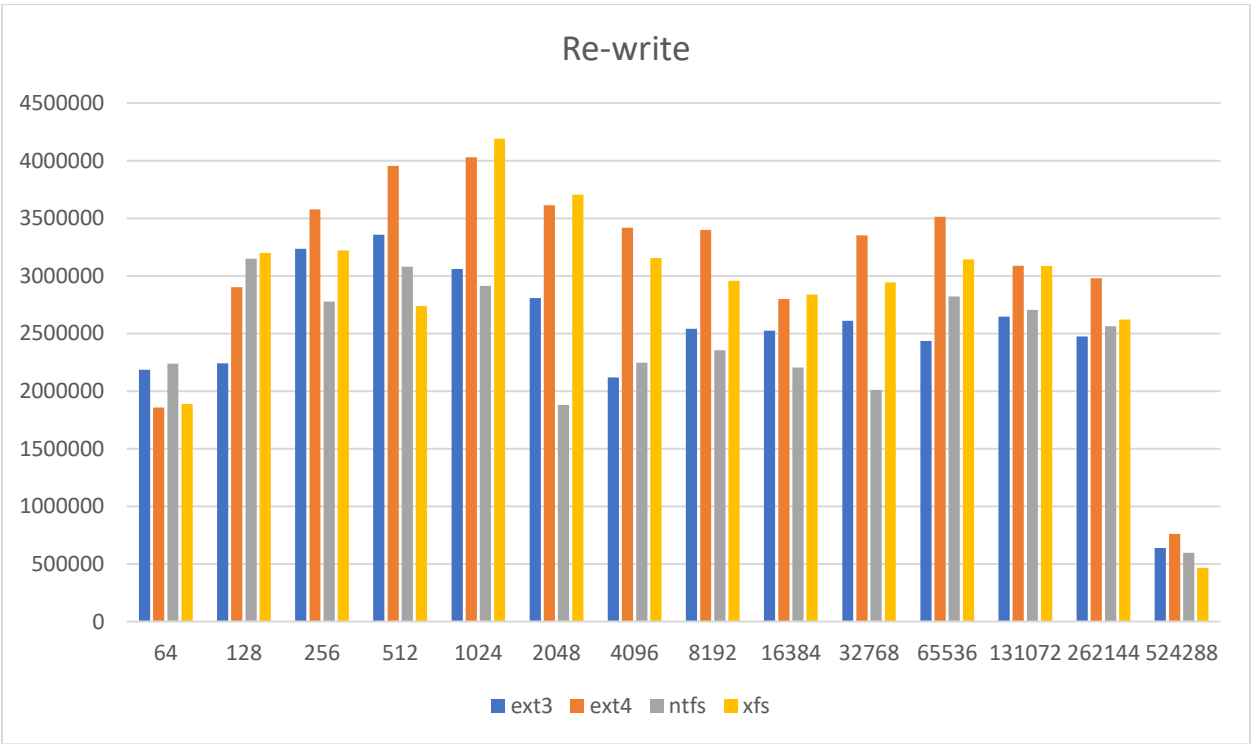
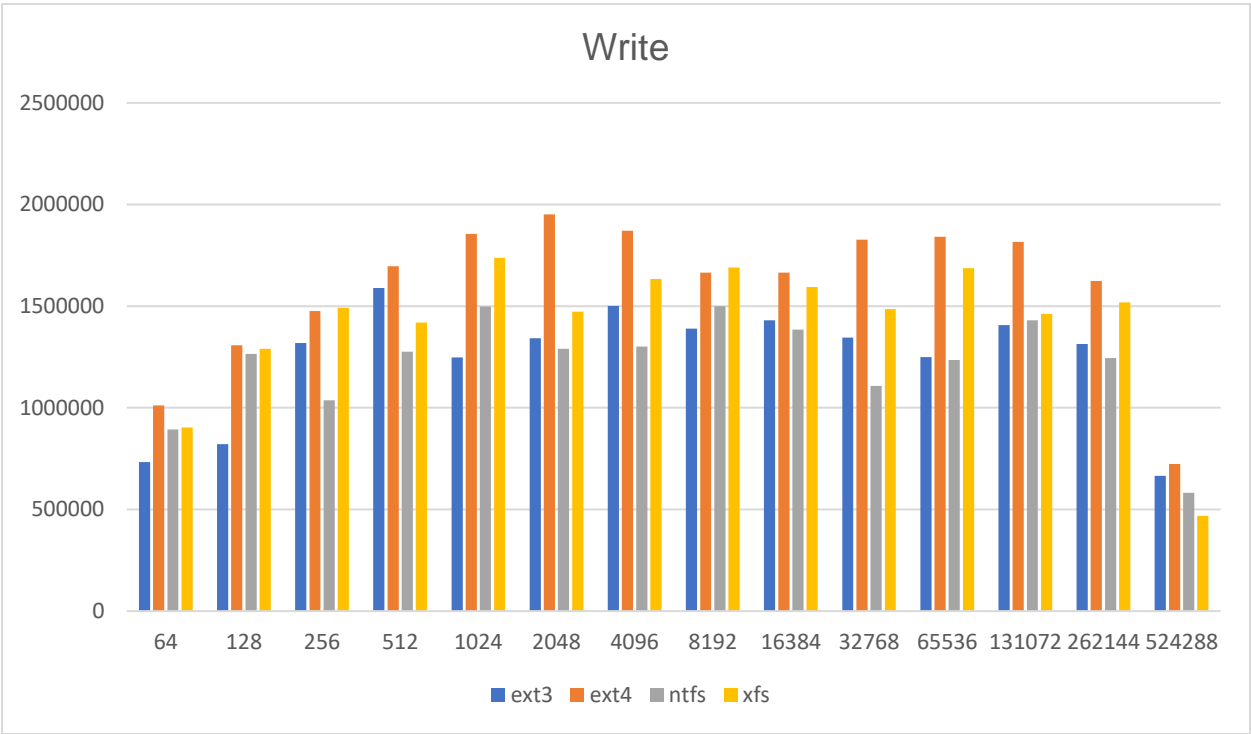
```
(huan@kali)-[~]
$ mkfs -t xfs -f xfs.bin
meta-data=xfs.bin      isize=512    agcount=4, agsize=32768 blks
                        =               sectsz=512   attr=2, projid32bit=1
                        =               crc=1        finobt=1, sparse=1, rmapbt=0
                        =               reflink=1     bigtime=1 inobtcount=1
data                  =               bsize=4096    blocks=131072, imaxpct=25
                        =               sunit=0      swidth=0 blks
naming                =version 2        bsize=4096    ascii-ci=0, ftype=1
log                   =internal log     bsize=4096    blocks=16384, version=2
                        =               sectsz=512   sunit=0 blks, lazy-count=1
realtime              =none             extsz=4096    blocks=0, rtextents=0
```

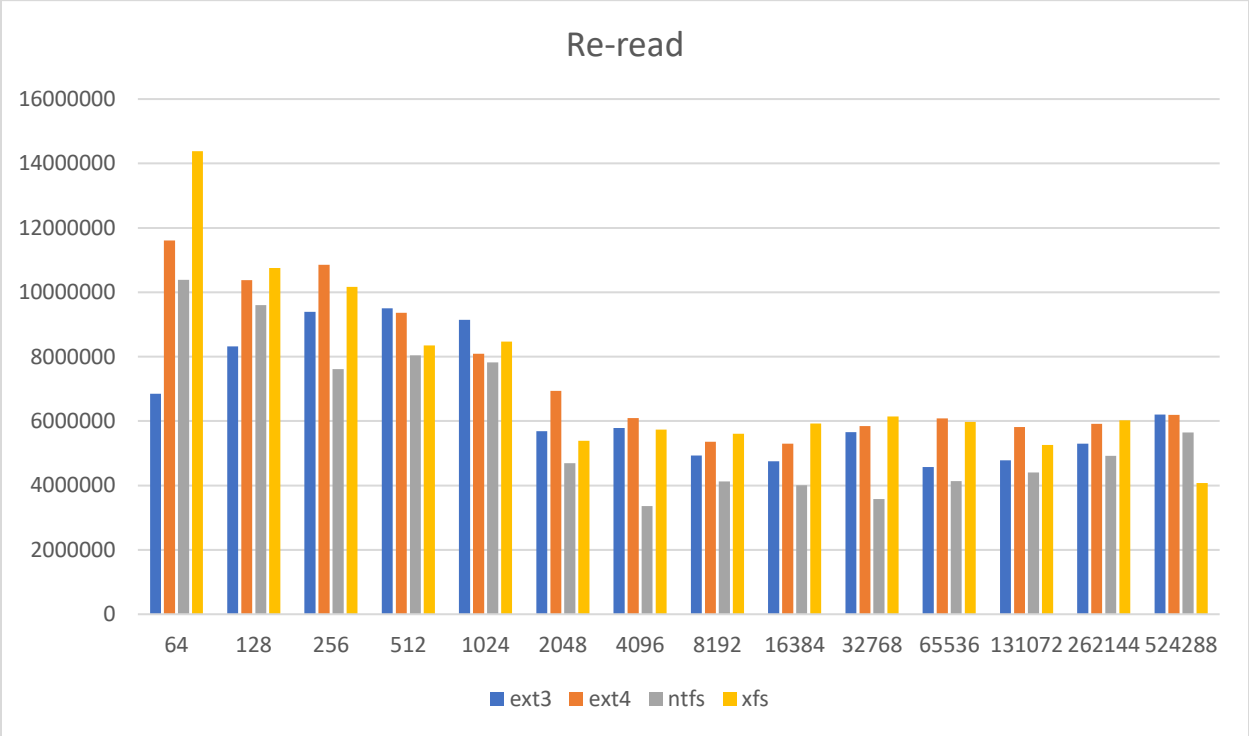
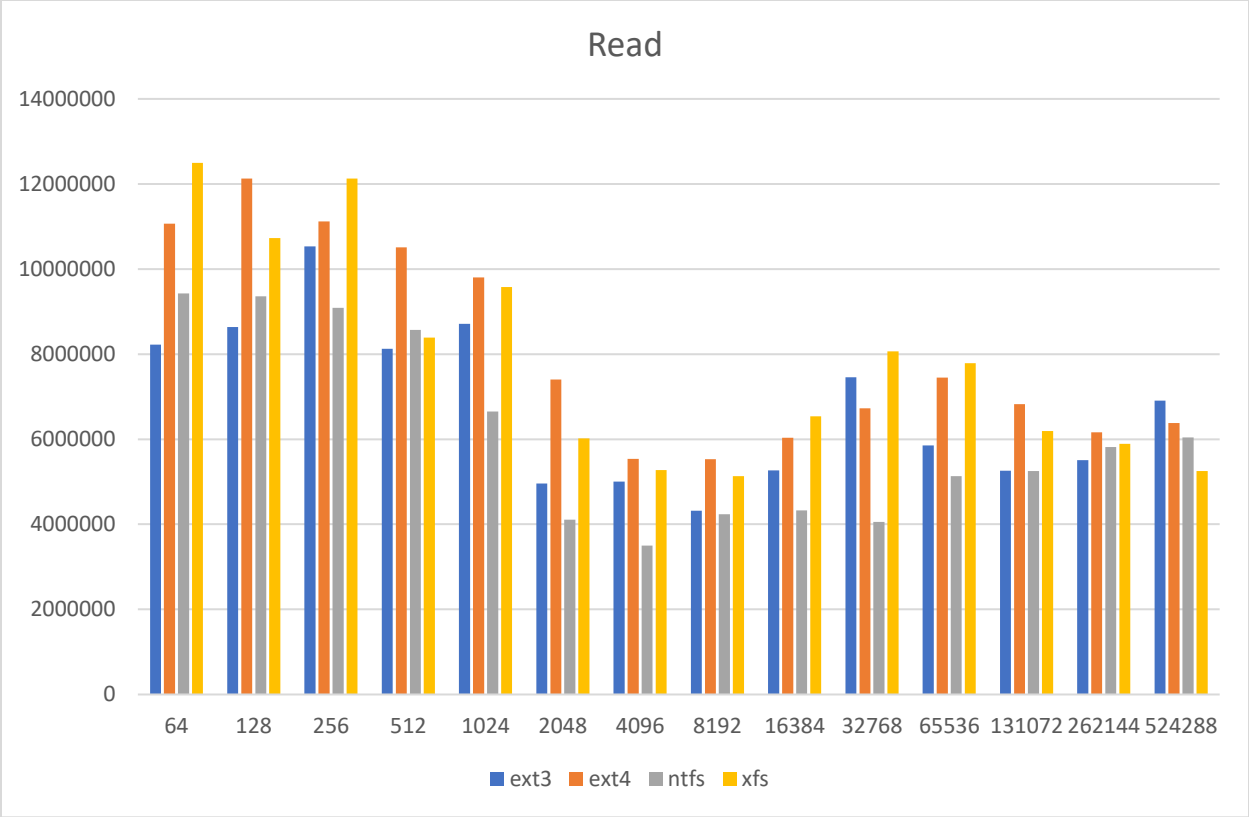
```
(huan@kali)-[~]
$ sudo mkdir /home/huan/OS/LAB5/xfs
```

```
(huan@kali)-[~]
$ sudo mount xfs.bin /home/huan/OS/LAB5/xfs
```

```
(huan@kali)-[~]
$ iozone -A /home/huan/OS/LAB5/xfs -b xfs.xls | tee xfs.txt
Iozone: Performance Test of File I/O
Version $Revision: 3.489 $
Compiled for 64 bit mode.
Build: linux-AMD64
```

III. Обработка данных и построение графиков





#### **IV. Вывод**

После выполнения лабораторной работы я обнаружил, что ext4 лучше всего подходит для моего компьютера. Помимо ext4, XFS также очень подходит для моего компьютера.