

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

Факультет безопасности информационных технологий

**Дисциплина:
«Операционные системы»**

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5
Тестирование файловых систем

Выполнила:
Студентка гр. №N3253
Пастухова А.А.



Проверил:
Ханов А.Р.

Санкт-Петербург
2022 г.

Задачи:

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

Усложненный вариант

Экзотическая файловая система

Ход работы:

Описание файловых систем, исследуемых в работе, 3 стандартных и 1 экзотическая:

FAT (таблица размещения файлов, file allocation table) – классическая архитектура файловой системы, которая из-за своей простоты всё ещё широко применяется для флеш-накопителей. Используется в дискетах, картах памяти и некоторых других носителях информации. Ранее находила применение и на жёстких дисках. Дисковое пространство логического раздела делится на две области – системную и область данных. Системная область создается и инициализируется при форматировании, а впоследствии обновляется при манипулировании файловой структурой. Область данных логического диска содержит файлы и каталоги, подчиненные корневому, и разделена на участки одинакового размера, минимальные единицы – кластеры. Кластер может состоять из одного или нескольких последовательно расположенных на диске секторов.

EXT2 (Extended File System) была разработана специально для операционной системы Linux. Это была первая файловая система по умолчанию в нескольких дистрибутивах Linux, таких как RedHat и Debian. Главной целью, которую преследовали создатели EXT было преодоление максимального размера записываемых файлов, который в то время составлял всего лишь 64 МБ. Благодаря созданию новой структуры метаданных – максимально возможный размер файла увеличился до 2 ГБ. В то же время максимальная длина имен файлов увеличилась с 255 байт. Она используется на обычных флэш-накопителях, таких как USB-накопитель, SD-карта и т. д. Главный недостаток ext2 (и одна из причин демонстрации столь высокой производительности) заключается в том, что она не является журналируемой файловой системой. Если во время записи файлов на диск случались сбои питания – файловая система повреждалась и восстановить ее было достаточно сложно.

BFS (Be File System) — файловая система, созданная для операционной системы BeOS. Это 64-битная журналируемая файловая система с поддержкой расширенных файловых атрибутов (метаданных), индексируемая, что приближает её функциональность к реляционным БД. Система может использоваться для разбивки дисков, CD-ROM, винчестеров и

флэш-носителей, хотя использование системы на небольших носителях проблематично: заголовки самой системы занимают от 600KB до 2MB.

ZFS (Zettabyte File System) продвинутая файловая система, имеющая некоторые особенности: объединенное хранилище (Pooled storage), copy-on-write, снапшоты, проверка целостности информации и автоматическая починка, RAID-Z, максимальный размер файла в 16 Эксабайт, максимальный размер хранилища в 256 квадрильонов Зеттабайт. Первое означает, что ZFS может создать файловую систему охватив все диски, а также добавить хранилище в систему дисков. Copy-on-write это другая интересная особенность - информация записывается в отдельный блок. Как только запись завершена, метаданные файловой системы обновляются к точке новой информации. Это гарантирует то, что, если файловая система сломается (или случится что-либо подобное) во время записи, старая информация будет фиксирована. Эта функция приводит к следующей вещи – снапшоты. ZFS использует снимки, для того чтобы следить за изменениями в файловой системе. Следующий факт можно объяснить тем, что всякий раз, когда новая информация записывается в ZFS, создается чек-сумма (контрольная сумма) для этой информации. Когда информация прочитана, чек-сумма подтверждается.

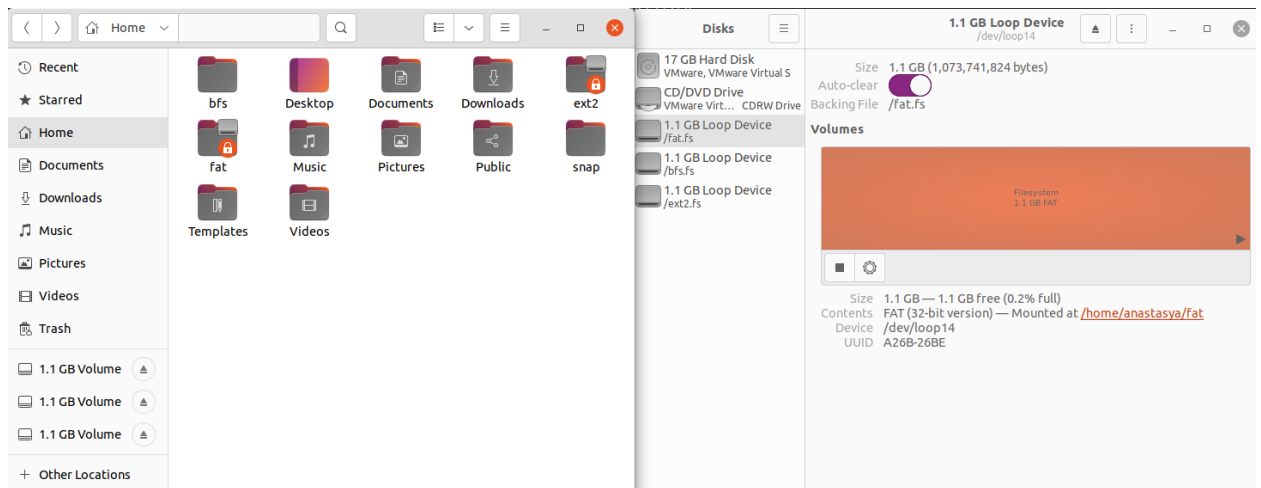
С помощью команды `mkfs` и двойного Tab я узнала доступные моей ОС файловые системы.

```
anastasya@ubuntu:/$ mkfs
mkfs          mkfs.cramfs  mkfs.ext3     mkfs.fat      mkfs.msdos    mkfs.vfat
mkfs.bfs      mkfs.ext2    mkfs.ext4     mkfs.minix    mkfs.ntfs
```

И поочередно создавала и монтировала файловые системы, следующими командами:

```
anastasya@ubuntu:/$ sudo dd if=/dev/zero of=/bfs.fs bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 2.82169 s, 381 MB/s
anastasya@ubuntu:/$ sudo mkfs.bfs /bfs.fs
anastasya@ubuntu:/$ sudo mount bfs.fs /home/anastasya/bfs
anastasya@ubuntu:/$
```

Проверила, что все они корректно установились на диске:



Затем я приступила к установке специфической файловой системы `zfs`.
Создала пул на основе одного диска `zfs.fs` и создала 3 базовых раздела:

`Pool0/data`

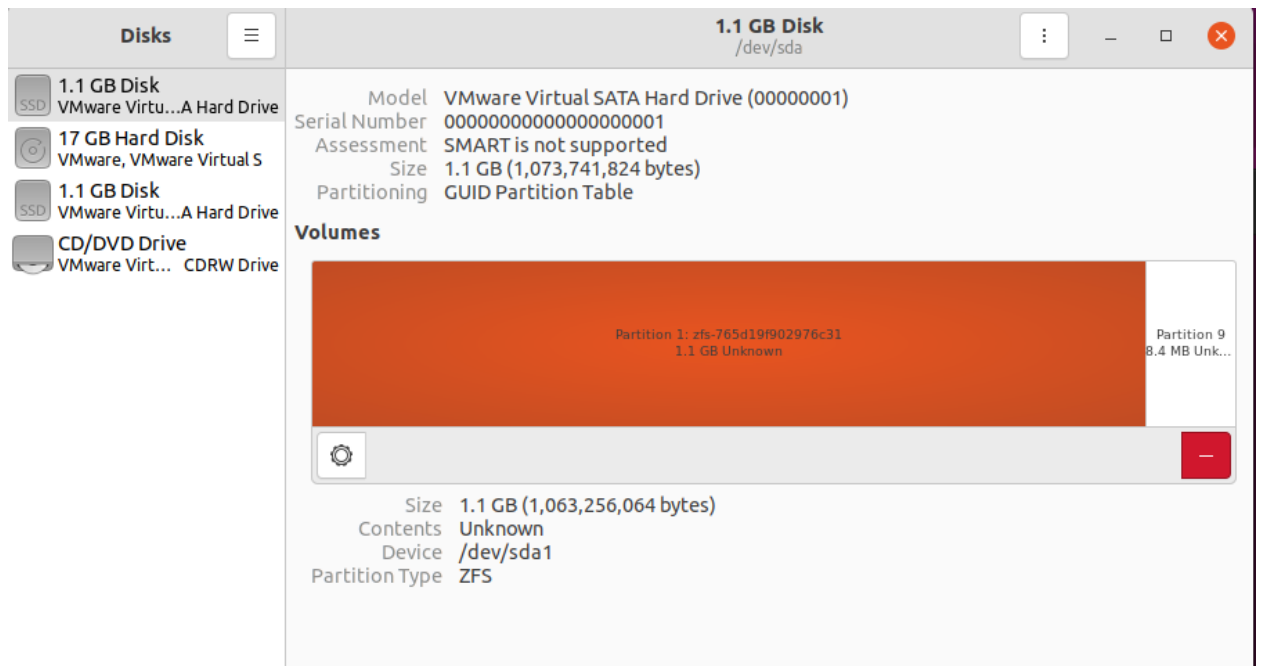
`Pool0/files`

`Pool0/media`

Далее пул был вмонтирован в систему как фс:

```
anastasya@ubuntu:/$ sudo zfs create pool0/data
anastasya@ubuntu:/$ sudo zfs create pool0/files
anastasya@ubuntu:/$ sudo zfs create pool0/media
anastasya@ubuntu:/$ sudo zfs mount
pool0                                /pool0
pool0/data                          /pool0/data
pool0/files                         /pool0/files
pool0/media                         /pool0/media
```

```
anastasya@ubuntu:/$ sudo dd if=/dev/zero of=/zfs.fs bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 3.33628 s, 322 MB/s
anastasya@ubuntu:/$ sudo zpool create -f pool0 /zfs.fs
anastasya@ubuntu:/$ sudo zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP    HEALTH  AL
TROOT
pool0     960M   104K   960M        -          -     0%    0%   1.00x   ONLINE  -
anastasya@ubuntu:/$
```



Проверила статус дисковых пространств и узнала, как называются именуются файловые системы:

```
anastasya@ubuntu:/$ df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           195M  1.7M  193M   1% /run
/dev/sdb3       16G   12G  2.7G  82% /
tmpfs           973M    0  973M   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sdb2       512M  5.3M  507M   2% /boot/efi
tmpfs           195M   96K  195M   1% /run/user/1000
pool0           832M  128K  832M   1% /pool0
pool0/data      832M  128K  832M   1% /pool0/data
pool0/files     832M  128K  832M   1% /pool0/files
pool0/media     832M  128K  832M   1% /pool0/media
```

Тестирование я проводила при помощи утилиты iotop, которая позволит проверить скорость работы файловых систем в различных режимах.

```

anastasya@ubuntu:/$ sudo iotest -A pool0
Iotest: 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: Thu Jun 16 05:52:46 2022

Auto Mode 2. This option is obsolete. Use -az -i0 -i1
Command line used: iotest -A pool0
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.

  kB  reflen  write  rewrite  read  reread
  64    4    388021  710511 2006158 1991276
  64    8    294723  939223 1828508 1780008
  64   16    415659 1083249 1780008 1638743
  64   32    278227  821391 1679761 1392258
  64   64    435202  811459 2298136 5860307

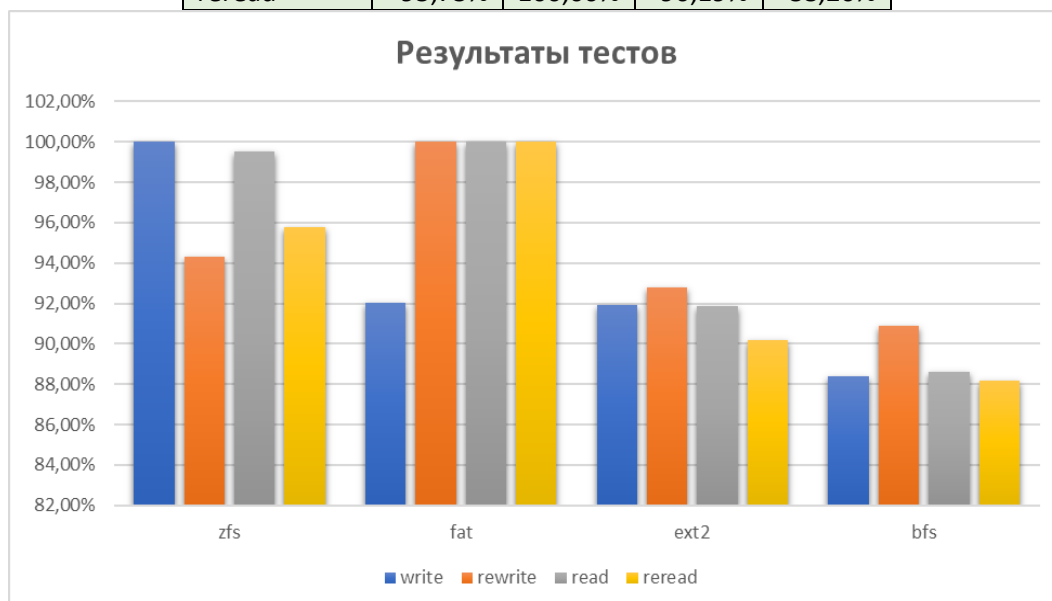
```

Получила такие результаты тестов, скорость в Kb/s

	zfs	fat	ext2	bfs
write	549976,9	506225,9	505473,5	486246
rewrite	1191133	1262937	1172093	1147999
read	3085691	3100724	2848859	2747521
reread	2478185	2588188	2334189	2282731

Для наглядного представления и построения графиков перевела отношение процент от лучшего результата.

в процент от лучшего	zfs	fat	ext2	bfs
write	100,00%	92,04%	91,91%	88,41%
rewrite	94,31%	100,00%	92,81%	90,90%
read	99,52%	100,00%	91,88%	88,61%
reread	95,75%	100,00%	90,19%	88,20%



Видно, что fat лидирует среди исследуемых файловых систем почти по всем показателям. Однако лишь экзотическая zfs система выдала наилучший результат записи, и достаточно хороший – для чтения. Скорость повторного чтения превышает первоначальную в системах fat, ext2, bfs, но этого нельзя сказать о повторной записи.

В моем личном рейтинге первое место занимает файловая система fat, так как она является классической и не имеет множество дополнительных фич, что обеспечивает высокую скорость чтения/записи.

Помощь и консультации в выполнении работы оказывал **Шарифуллин И.А.**