ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет безопасности информационных технологий

Дисциплина:
“Операционные системы”

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

Выполнил:
Студент гр. N3249
Шарифуллин Ильдан Айдарович

Проверил:
Савков С.В.

Санкт-Петербург
2022г.

Задание:

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

Усложненный вариант

Экзотические фс или Экзотические методики проверки

Ход работы:

В ходе работы я буду тестировать 4 файловые системы. 2 из них будут экзотическими и 2 стандартных (для сравнения). Я выбрал:

minix – Первоначально создано Эндрю С. Таненбаум как учебное пособие, МИНИКС представляет собой операционную систему «mini-Unix». В настоящее время он нацелен на создание самовосстанавливающейся и отказоустойчивой операционной системы. Файловая система MINIX была разработана как упрощенная версия файловой системы Unix. Экзотической она является еще и потому, что максимальный ее объем равен 64мб.

vfat – Таблица размещения виртуальных файлов, была введена в Windows 95 и сняла ограничение на восемь символов для имен файлов. Стало возможным имена файлов длиной до 255 символов. Единственная причина, по которой вы будете использовать эту файловую систему, — это совместимость с операционными системами, отличными от Linux.

bfs – Это Загрузочная файловая система, который предназначен для одной и только одной задачи: для обработки файлов в загрузочном разделе. Редко приходится создавать загрузочную файловую систему вручную ведь обычно это делает сам Linux. Экзотическая она в данном случае потому, как мы будем тестировать ее в качестве стандартной файловой системы.

ext4 – это стандартная файловая система для некоторых дистрибутивов Linux. Это надежная, проверенная и надежная файловая система. Он имеет функции, которые уменьшить фрагментацию файлов и может использоваться с более крупными дисками, разделами и файлами, чем Ext3.

Для тестирования я буду использовать утилиту iozone, которая позволит проверить мне скорость работы файловых систем в различных режимах.

С помощью следующих команд я создавал файловые системы и монтировал их на свою ОС:

```
sudo dd if=/dev/zero of=/file2.fs bs=1M count=64
sudo mkfs.vfat /file2.fs
sudo mount — t minix — o loop file2.fs /home/ildan/fs2
```

Стоит отметить, что ФС minix имеет ограничение в 64Мб, из-за чего все тесты были проведены на файлах именно такого размера.

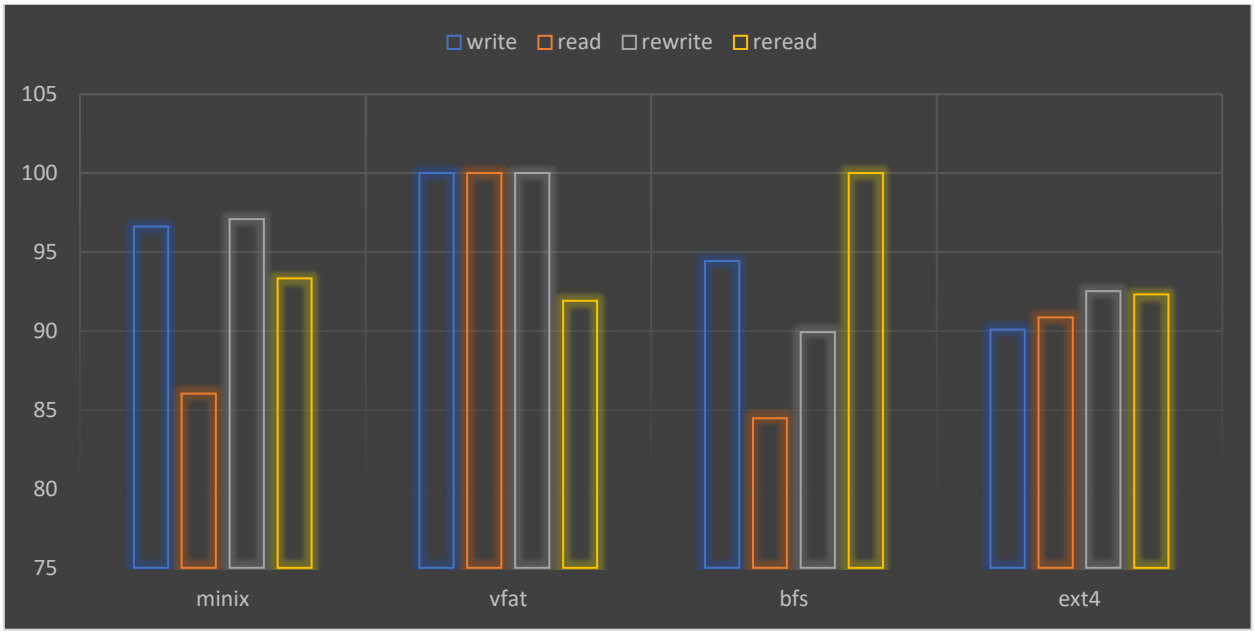
В результате тестов я получил следующие результаты:

Kb/s	write	read	rewrite	reread
minix	801372,9	2176588	1771094	1793610
vfat	829355,2	2529762	1824093	1766361
bfs	783178,7	2137071	1640572	1921620
ext4	747176,4	2298802	1687731	1774053

В процентах от лучшего варианта:

%	write	read	rewrite	reread
minix	96,62602	86,03924	97,09449	93,33841
vfat	100	100	100	91,9204
bfs	94,43224	84,47716	89,93904	100
ext4	90,09124	90,87028	92,52438	92,32067

Из чего можно сделать вывод о том, что ФС vfat выдает лучшие показатели скорости среди выбранных мною ФС. Посмотрим на те же данные на графике:



Несмотря на свою полную непредусмотренность ФС minix смогла обойти другие ФС по некоторым параметрам. Поскольку тесты проводились на файлах объемом 64Мб, эти тесты сложно считать реальными и достоверными.

На этот раз я возьму только ФС, которые реально предназначены для обычного пользовательского опыта. Объем файлов на этот раз тоже будет максимально реальным. Тесты проводились с помощью все той же утилиты iozone.
Параметры тестов:
Объем: 64kb – 512 mb
Длина записи/чтения (сколько памяти запрашивается при каждом системном вызове): 4kb – 16384kb (если объем файла меньше, то ограничивался длинной, равной этому объему)

Тестируемые ФС:
BTRFS - Современная ФС, главной особенностью которой является высокая отказоустойчивость. Из дополнительных «бонусов»: удобна для сисадминов и поддерживает сравнительно простой процесс восстановления данных. Поддерживает подтома, разрешает менять размеры разделов в динамическом режиме и позволяет делать снапшоты. Отличается высокой производительность. Применяется как ФС, установленная по умолчанию, в OpenSUSE и SUSE Linux. Главный минус — нестабильность (нарушена обратная совместимость, сложная для поддержки и так далее).

ZFS — новейшая транзакционная файловая система, работающая с использованием механизма copy-on-write и оперирующая одновременно на нескольких уровнях абстракции данных.

И уже упомянутая **ext4.**

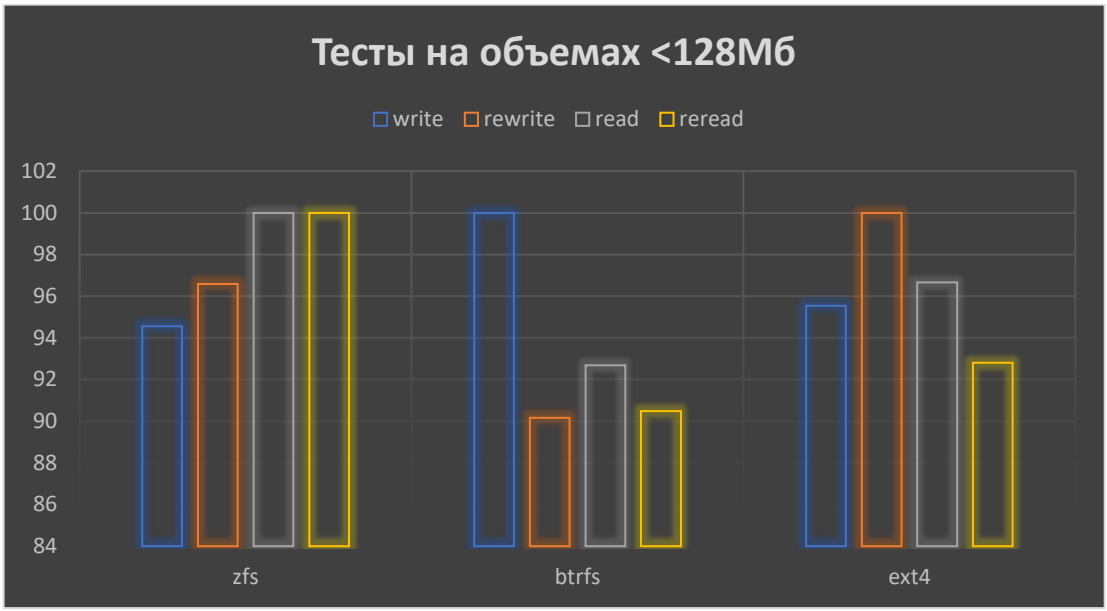
Все эти ФС являются журналируемыми, поэтому результаты тестов должны быть максимально объективными.
Поскольку ZFS по умолчанию не включена в большую часть дистрибутивов Linux, пришлось устанавливать ее отдельно. Скриншот установки приведен ниже:

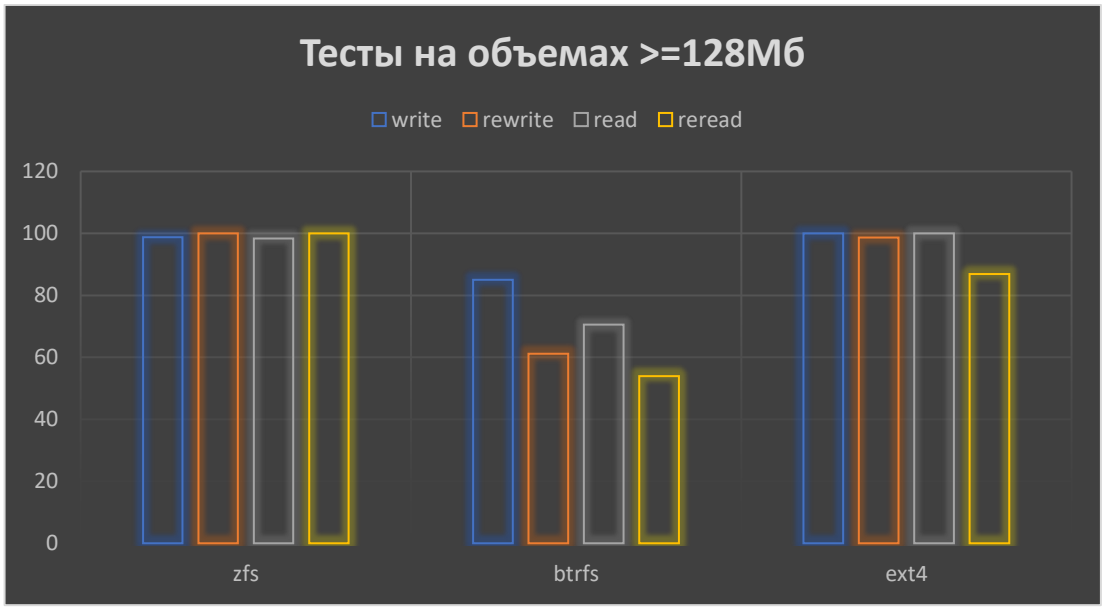
```
ildan@vb:/$ sudo lsblk -S
NAME HCTL       TYPE VENDOR   MODEL            REV SERIAL                TRAN
sda  2:0:0:0    disk ATA      VBOX_HARDDISK 1.0  VBcb87d3b6-073042b1 sata
sdb  3:0:0:0    disk ATA      VBOX_HARDDISK 1.0  VBac2127e4-9b25607e sata
sdc  4:0:0:0    disk ATA      VBOX_HARDDISK 1.0  VB72d0e5af-38f1a7c7 sata
sr0  1:0:0:0    rom  VBOX     VBOX_CD-ROM   1.0  VB2-01700376        ata
ildan@vb:/$ sudo zpool create -f fs_z /dev/sdb /dev/sdc
ildan@vb:/$ zpool status
  pool: fs_z
 state: ONLINE
config:

        NAME        STATE     READ WRITE CKSUM
        fs_z        ONLINE       0     0     0
          sdb       ONLINE       0     0     0
          sdc       ONLINE       0     0     0

errors: No known data errors
```

Результаты тестов:





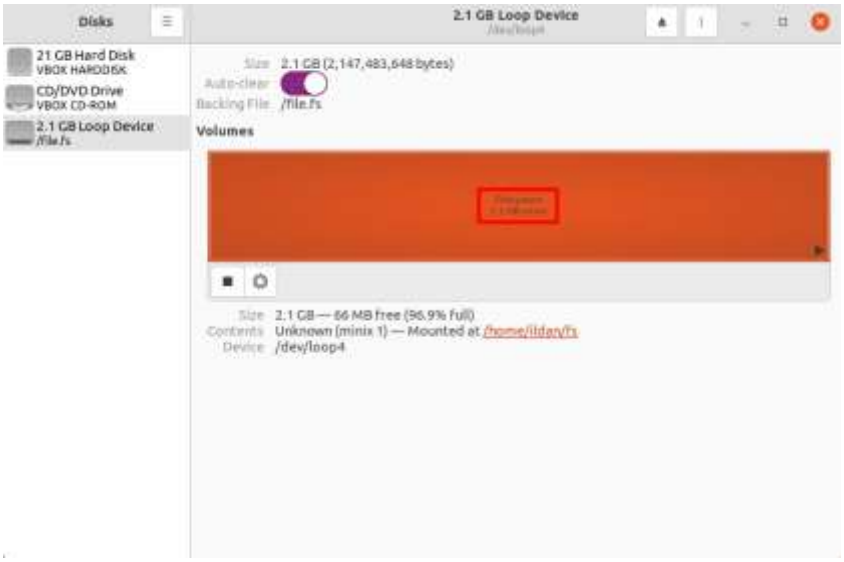
Увидев результаты, я очень обрадовался тому, что решил разделить тесты на 2 категории. Файлы объемом до 128Мб и больше либо равные данного объема. Результаты очень сильно разнятся. Начнем разбор результатов со второй категории. На графиках видно, что с большими файлами одинаково хорошо справились ZFS и ext4, а вот результаты btrfs выглядят очень грустно на фоне упомянутых ФС. Странно видеть такие результаты с учетом довольно громких слов об этой ФС. Я решил почитать различные сравнения и обсуждение на тему этого, и действительно, на практике у всех btrfs работает медленнее. И дело тут даже не в тех фичах, которые нам предлагают его разработчики. Дело в том, что данная система особо не поддерживается уже долгое время, и до сих считается нестабильной. На малом количестве тестов она может себя неплохо показать, но в среднем всегда будет проигрывать ext4. Если сравнивать ZFS и ext4, то можно увидеть, что все результаты кроме повторного чтения равны. Это объясняется тем, что в ZFS есть хорошо и стабильно (в отличие от btrfs) прописанная функция Copy-On-Write. Ее суть в том, что при чтении области данных используется общая копия, в случае изменения данных — создается новая копия.

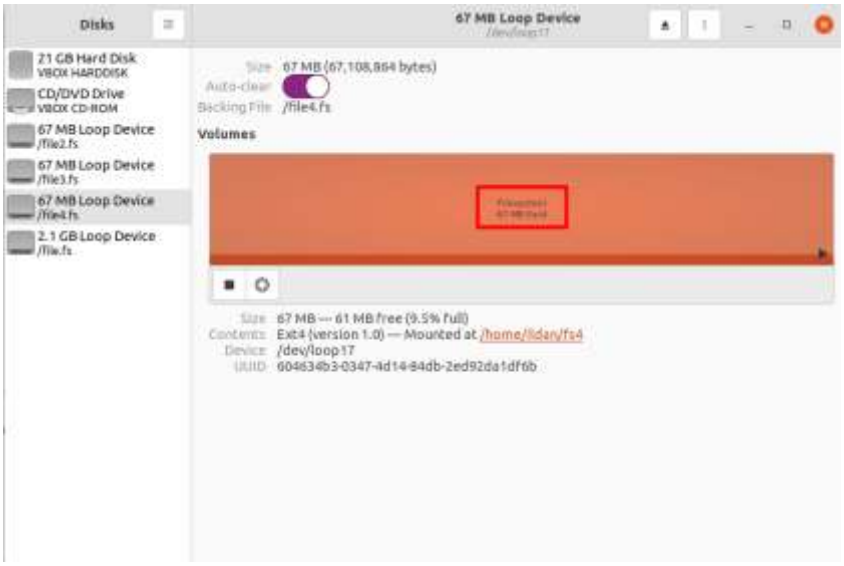
Перейдем к первой категории тестов – файлы маленького объема. Btrfs по-прежнему занимает последнее место в рейтинге и все по той же причине. Здесь интереснее рассмотреть разницу результатов оставшихся двух ФС. Здесь все также видна полезность вышеупомянутой COW, из-за которой повторное чтение и запись в ZFS происходит быстрее (на 3-7%), хоть эта разница и не так впечатляет, как в первой категории (14% на повторное чтение). А вот что касается простых чтения и записи, лидирует надежная и стабильная ext4. Почитав про разницу в работе этих ФС я понял, что на деле, ext4 является абсолютно универсальной, а ZFS все же несколько специализированной ФС. На практике ZFS показывает себя хорошо на больших, серверных объемах данных. Но все же и на пользовательском объеме проигрывает незначительно.

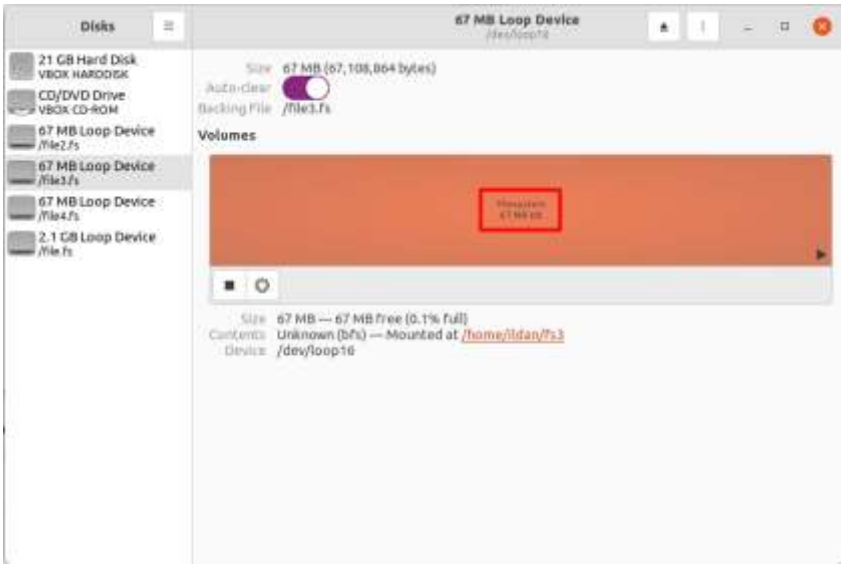
Вывод:

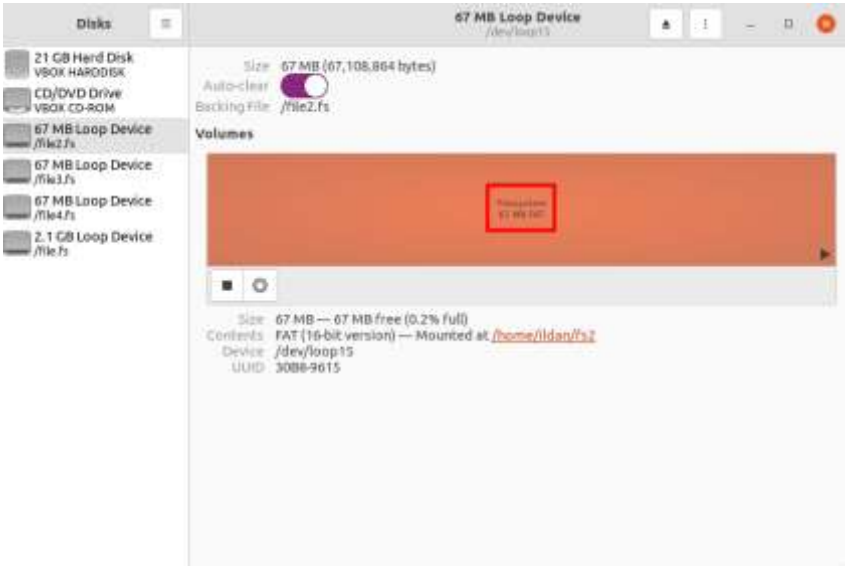
В ходе экспериментов я выяснил, что в реальных условиях (пользовательский опыт), хуже всего справляется btrfs из-за своей нестабильности. Лучше справляется с первичной записью и чтением ext4 за счет отсутствия большого количества фич для специализированных операций, постоянной поддержки, хорошей стабильности. А с повторной записью и чтением отлично справляется ZFS, за счет технологии copy-on-write.

Приложение:


Disks
2.1 GB Loop Device
21 GB Hard Disk
VBOX HARDDISK
CD/DVD Drive
VBOX CD-ROM
2.1 GB Loop Device
/file.fs
Size 2.1 GB (2,147,483,648 bytes)
Auto-clear
Backing File /file.fs
Volumes
Size 2.1 GB — 66 MB free (96.9% full)
Contents Unknown (minix 1) — Mounted at /home/ildar/fs
Device /dev/loop4


Disks
67 MB Loop Device
21 GB Hard Disk
VBOX HARDDISK
CD/DVD Drive
VBOX CD-ROM
67 MB Loop Device
/file2.fs
67 MB Loop Device
/file3.fs
67 MB Loop Device
/file4.fs
2.1 GB Loop Device
/file.fs
Size 67 MB (67,108,864 bytes)
Auto-clear
Backing File /file4.fs
Volumes
Size 67 MB — 61 MB free (9.5% full)
Contents Ext4 (version 1.0) — Mounted at /home/ildar/fs4
Device /dev/loop17
UUID 604634b3-0347-4d14-84db-2ed92da1df6b


Disks
67 MB Loop Device
21 GB Hard Disk
VBOX HARDDISK
CD/DVD Drive
VBOX CD-ROM
67 MB Loop Device
/file2.fs
67 MB Loop Device
/file3.fs
67 MB Loop Device
/file4.fs
2.1 GB Loop Device
/file.fs
Size 67 MB (67,108,864 bytes)
Auto-clear
Backing File /file3.fs
Volumes
Size 67 MB — 67 MB free (0.1% full)
Contents Unknown (bfs) — Mounted at /home/ildar/fs3
Device /dev/loop16




```
Run began: Wed Apr 13 03:25:24 2022

File size set to 65536 kB
Auto Mode
Command line used: iozone -s 65536 -a /dev/loop17
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.
                                                            random    random     bkwd   record   stride
      kB  reclen    write  rewrite    read    reread    read     write     read  rewrite     read   fwrite frewrite   fread  freread
   65536       4   365420   856163 1230331   891572  701587   516361   780237   686584   712045   653867   402451 1033104 1189315
   65536       8   441053  1117695 1511681  1174502  983699   792665  1388210  1284189  1129368   791658  1998686 1696287 1455768
   65536      16   514738  1951512 1878674  1784552 1828012   728234  1862689  1592767  2065038   998598  1708177 2879454 2041109
   65536      32   471624  2445558 4295517  1603012 2371583   903357  2186582  3035708  1743719   848073  2334225 3213038 2655111
   65536      64   446935  2911817 3647007  1977370 2819960  1330678  4171338  1953911  2449110  1009597  2487201 4217225 2435720
   65536     128   490782  2677862 2617366  1595291 2934885  1048104  3951279  2288461  2451338  1094268  2178582 2980969 2216832
   65536     256   520089  3007916 3453422  1888198 2277840  1096188  3113641  2178289  2892989   826430  1997255 4507539 2392097
   65536     512   553640  2158973 4293772  1820943 3874669   932792  3917645  2888584  1780984   862498  1821254 4782082 2937928
   65536    1024   396766  2021223 3792580  1832558 2545193   945823  2014764  1576597  2246160   780043   520391 2134927 1620534
   65536    2048   317041   815277  857857  1008588 1316750   589925  1522575   956086  2438356   895275   402882 2822532 2237785
   65536    4096   376278  1106523 1928334  1342933 2635055  1060404  1961748  1455614  1474375   987927  1485602 3985424 1883166
   65536    8192   371165  2154725 1856282  1457134 1731789  1212979  2248776  1866423  1548356   898616  1442226 1819857 1882494
   65536   16384   285450  1572098 1850440  1320680  731673  1184758  1340123   915184  1586124   577389  2094470 1833585 2094342
```