1МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»
ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра компьютерной инженерии и моделирования

ОТЧЕТ ПО ПРАКТИЧЕСКОМУ ЗАДАНИЮ №7
«Файловые системы операционных систем»

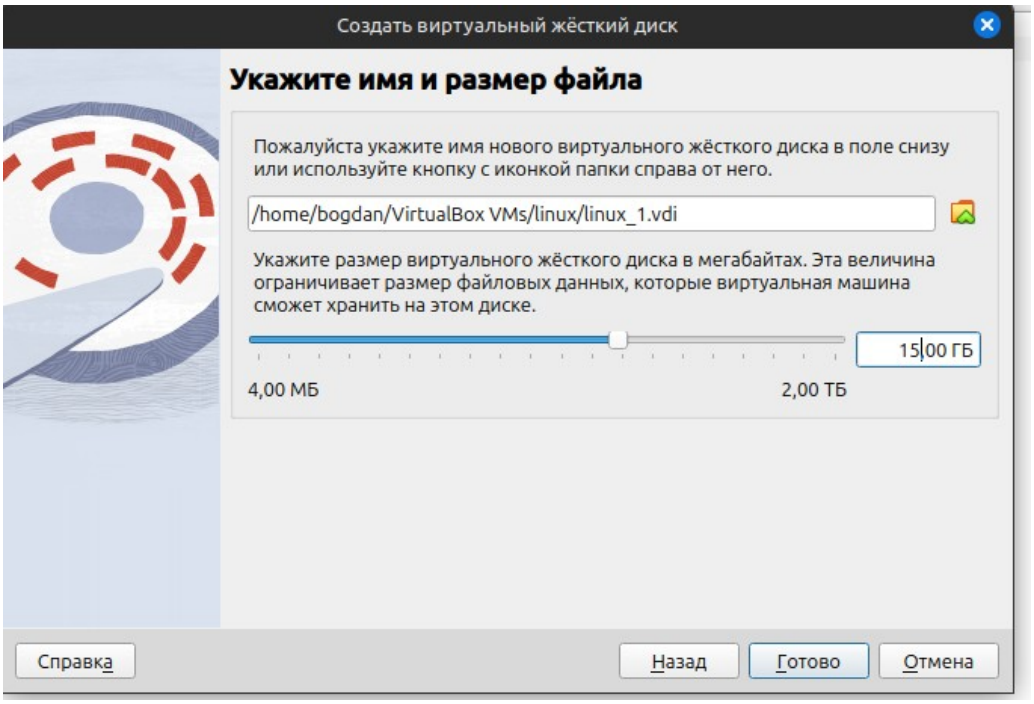
Практическая работа
по дисциплине «Системное программное обеспечение»
студента 3 курса группы ИВТ-б-о-222(2)
Чудопалова Богдана Андреевича

09.03.01 «Информатика и вычислительная техника»

Симферополь, 2025

Ход работы

1. Создать в виртуальной машине неразмеченный диск, который затем использовать для создания различных файловых систем.



Структура диска

```
Создан новый раздел 10 с типом 'Linux filesystem' и размером 4 GiB.

Команда (m для справки): p
Диск /dev/sdb: 15 GiB, 16106127360 байт, 31457280 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: gpt
Идентификатор диска: 8359DAA0-6DC5-41BA-A54D-99563D5EE352

Устр-во       начало    Конец Секторы Размер Тип
/dev/sdb1       2048  2099199 2097152     1G Файловая система Linux
/dev/sdb2    2099200  4196351 2097152     1G Файловая система Linux
/dev/sdb3    4196352  6293503 2097152     1G Файловая система Linux
/dev/sdb4    6293504  8390655 2097152     1G Файловая система Linux
/dev/sdb5    8390656 10487807 2097152     1G Файловая система Linux
/dev/sdb6   10487808 12584959 2097152     1G Файловая система Linux
/dev/sdb7   12584960 14682111 2097152     1G Файловая система Linux
/dev/sdb8   14682112 16779263 2097152     1G Файловая система Linux
/dev/sdb9   16779264 18876415 2097152     1G Файловая система Linux
/dev/sdb10 18876416 27265023 8388608     4G Файловая система Linux

Команда (m для справки):
```

Далее начал создание необходимых файловых систем

```
bogdan@bogdan-virtualbox:~$ sudo mkfs.ext2 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 262144 4k blocks and 65536 inodes
UUID файловой системы: e0946cea-fd10-418e-af1b-eca7abe2ca89
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Распределение групповых таблиц: готово
Сохранение таблицы inod'ов: готово
Writing superblocks and filesystem accounting information: готово

bogdan@bogdan-virtualbox:~$
```

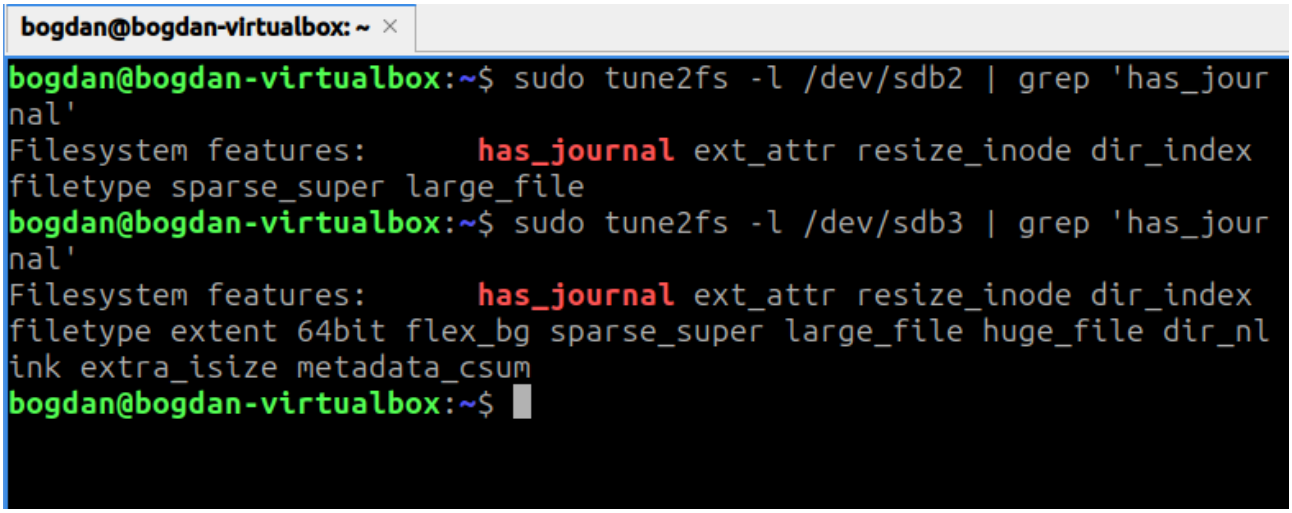
Создание FAT32 и reiserfs

```
bogdan@bogdan-virtualbox:~$ sudo mkfs.reiserfs /dev/sdb7
mkfs.reiserfs 3.6.27

Guessing about desired format.. Kernel 6.11.0-17-generic is running.
Format 3.6 with standard journal
Count of blocks on the device: 262144
Number of blocks consumed by mkreiserfs formatting process: 8219
Blocksize: 4096
Hash function used to sort names: "r5"
Journal Size 8193 blocks (first block 18)
Journal Max transaction length 1024
inode generation number: 0
UUID: 27c5f8e4-b5ec-46be-823a-a2bc4ac9fbfd
ATTENTION: YOU SHOULD REBOOT AFTER FDISK!
        ALL DATA WILL BE LOST ON '/dev/sdb7'!
Continue (y/n):y
Initializing journal - 0%....20%....40%....60%....80%....100%
Syncing..ok
ReiserFS is successfully created on /dev/sdb7.
bogdan@bogdan-virtualbox:~$ sudo mkfs.vfat -F 32 /dev/sdb8
mkfs.fat 4.2 (2021-01-31)
bogdan@bogdan-virtualbox:~$
```

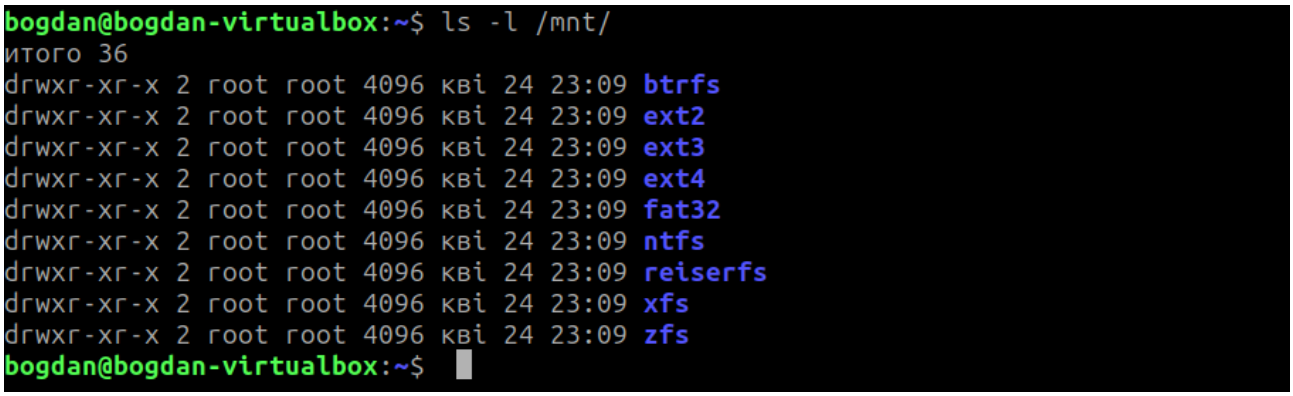
Создание ntfs

```
bogdan@bogdan-virtualbox:~$ sudo mkfs.ntfs /dev/sdb9
Cluster size has been automatically set to 4096 bytes.
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
bogdan@bogdan-virtualbox:~$ sudo mkswap /dev/sdb10
Setting up swapspace version 1, size = 4 GiB (4294963200 bytes)
без метки, UUID=ffff7e38-3ab2-42d3-b3f1-c02ac703f856
bogdan@bogdan-virtualbox:~$ sudo swapon /dev/sdb10
bogdan@bogdan-virtualbox:~$
```

Проверка опции журналирования



Итоговый вид



12. Настроить ОС, чтобы все файловые системы монтировались при старте системы — с помощью команды blkid узнал UUID каждого раздела

Далее перенес это в /etc/fstab

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a device; this may
# be used with UUID= as a more robust way to name devices that works even if
# disks are added and removed. See fstab(5).
#
# <file system>             <mount point>  <type>  <options>  <dump>  <pass>
UUID=739c6092-c851-4389-b592-42581b82c4e7 /              ext4    defaults   0 1
/swapfile                                 swap           swap    defaults   0 0

UUID=e0946cea-fd10-418e-af1b-eca7abe2ca89 /mnt/ext2      ext2    defaults   0 2
UUID=b3460241-05dd-4a27-b21f-e788b4aef4cf /mnt/ext3      ext3    defaults   0 2
UUID=a6e976eb-f6db-4e7d-9aae-f861145fdf3c /mnt/ext4      ext4    defaults   0 2
UUID=15d20612-a422-40ad-b8c1-00f0029a620c /mnt/xfs       xfs     defaults   0 2
UUID=d15c1f01-f4d0-4ff0-b5f8-80a7034f5ad6 /mnt/btrfs     btrfs   defaults   0 2
UUID=27c5f8e4-b5ec-46be-823a-a2bc4ac9fbfd /mnt/reiserfs  reiserfs defaults  0 2
UUID=883F-0A13                            /mnt/fat32     vfat    defaults   0 2
UUID=58973F026769D6A8                     /mnt/ntfs      ntfs    defaults   0 2
UUID=ffff7e38-3ab2-42d3-b3f1-c02ac703f856 none           swap    sw         0 0
~
```

15. Протестировать все разделы для операций чтения маленьких файлов (размер 16Кб), цикл тестирования не менее 100 раз — был создан скрипт

bogdan@bogdan-virtualbox: ~/scripts

```
  /mnt/ext4
  /mnt/xfs
  /mnt/btrfs
  /mnt/zfs
  /mnt/reiserfs
  /mnt/fat32
  /mnt/ntfs
)

for mp in "${mount_points[@]}"; do
  echo "Подготовка каталога test_16k в $mp"
  mkdir -p "$mp/test_16k"
done

for mp in "${mount_points[@]}"; do
  dd if=/dev/zero of="$mp/test_16k/file_16k" bs=16K count=1 oflag=direct 2>/dev/null
done

for mp in "${mount_points[@]}"; do
  echo "Выполнение теста чтения в $mp"
  start=$(date +%s.%N)
  for i in {1..100}; do
    dd if="$mp/test_16k/file_16k" of=/dev/null bs=16K count=1 iflag=direct 2>/dev/null
  done
  end=$(date +%s.%N)
  duration=$(echo "$end - $start" | bc)
  result="Шаг15: Время чтения 100 раз для $mp: $duration секунд"
  echo "$result" | tee -a "$RESULTS_FILE"
done
```

Получил следующие результаты:

```
Шаг15: Время чтения 100 раз для /mnt/ext2: .211437716 секунд
Шаг15: Время чтения 100 раз для /mnt/ext3: .212518345 секунд
Шаг15: Время чтения 100 раз для /mnt/ext4: .210519835 секунд
Шаг15: Время чтения 100 раз для /mnt/xfs: .212930673 секунд
Шаг15: Время чтения 100 раз для /mnt/btrfs: .219956943 секунд
Шаг15: Время чтения 100 раз для /mnt/zfs: .211293036 секунд
Шаг15: Время чтения 100 раз для /mnt/reiserfs: .207319939 секунд
Шаг15: Время чтения 100 раз для /mnt/fat32: .213182841 секунд
Шаг15: Время чтения 100 раз для /mnt/ntfs: .226434556 секунд
~
```

16. Протестировать все файловые системы для операций записи больших файлов (500Мб) — для этого создал bash скрипт

```
#!/bin/bash
RESULTS_FILE="16time_res.log"
mount_points=(
  /mnt/ext2
  /mnt/ext3
  /mnt/ext4
  /mnt/xfs
  /mnt/btrfs
  /mnt/zfs
  /mnt/reiserfs
  /mnt/fat32
  /mnt/ntfs
)
echo "=== Шаг 16: Последовательная запись файла 500 Мб ==="
for mp in "${mount_points[@]}"; do
  echo "Создание каталога largefile_test в $mp"
  mkdir -p "$mp/largefile_test"
done
for mp in "${mount_points[@]}"; do
  echo "Тест последовательной записи в $mp"
  start=$(date +%s.%N)
  dd if=/dev/zero of="$mp/largefile_test/largefile" bs=1M count=500 oflag=sync 2>&1 | tee -a /dev/null
  end=$(date +%s.%N)
  duration=$(echo "$end - $start" | bc)
  result="Шаг16: Время последовательной записи 500 Мб на $mp: $duration секунд"
  echo "$result" | tee -a "$RESULTS_FILE"
done
```

Получил следующий результат

```
Шаг16: Время последовательной записи 500 Мб на /mnt/ext2: 2.595074794 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/ext3: 3.093639995 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/ext4: 2.484476616 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/xfs: 1.683783160 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/btrfs: 2.587081145 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/zfs: 2.514645109 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/reiserfs: 18.275327761 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/fat32: 3.056239671 секунд
Шаг16: Время последовательной записи 500 Мб на /mnt/ntfs: 11.586481324 секунд
```

17. Протестировать операции записи для маленьких файлов — создал следующий скрипт

```
RESULTS_FILE="17time_res.log"
mount_points=(
  /mnt/ext2
  /mnt/ext3
  /mnt/ext4
  /mnt/xfs
  /mnt/btrfs
  /mnt/zfs
  /mnt/reiserfs
  /mnt/fat32
  /mnt/ntfs
)

echo "=== Шаг 17: Запись 1000 маленьких файлов (16КБ) ==="
for mp in "${mount_points[@]}"; do
  echo "Создание каталога small_files_test в $mp"
  mkdir -p "$mp/small_files_test"
done
for mp in "${mount_points[@]}"; do
  echo "Начало теста записи маленьких файлов в $mp"
  start=$(date +%s.%N)
  for i in $(seq 1 1000); do
    dd if=/dev/zero of="$mp/small_files_test/file_$i" bs=16K count=1 oflag=sync 2>/dev/null
  done
  end=$(date +%s.%N)
  duration=$(echo "$end - $start" | bc)
  result="Шаг17: Общее время записи 1000 файлов в $mp: $duration секунд"
  echo "$result" | tee -a "$RESULTS_FILE"
done
```

Получил следующий результат

```
Шаг17: Общее время записи 1000 файлов в /mnt/ext2: 3.368117121 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/ext3: 4.202304476 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/ext4: 4.229960443 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/xfs: 4.248050513 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/btrfs: 4.635441535 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/zfs: 4.124515784 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/reiserfs: 11.797354885 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/fat32: 5.467851021 секунд
Шаг17: Общее время записи 1000 файлов в /mnt/ntfs: 5.250404105 секунд
~
```

18. Протестировать операции записи для больших файлов — был создан следующий скрипт.

```
#!/bin/bash
RESULTS_FILE="18time_res.log"
mount_points=(
  /mnt/ext2
  /mnt/ext3
  /mnt/ext4
  /mnt/xfs
  /mnt/btrfs
  /mnt/zfs
  /mnt/reiserfs
  /mnt/fat32
  /mnt/ntfs
)

for mp in "${mount_points[@]}"; do
  echo "Запуск fio для записи на $mp" | tee -a "$RESULTS_FILE"
  fio --name=randwrite_test \
      --filename="$mp/largefile_rand" \
      --size=300M \
      --bs=1M \
      --rw=randwrite \
      --iodepth=16 \
      --numjobs=1 \
      --time_based \
      --runtime=60 \
      --direct=1 | tee -a "$RESULTS_FILE"
done
```

19. Создать структуру каталогов с не менее чем 1000 подкаталогов в каждой файловой системе, замерить время создания — был создан следующий скрипт

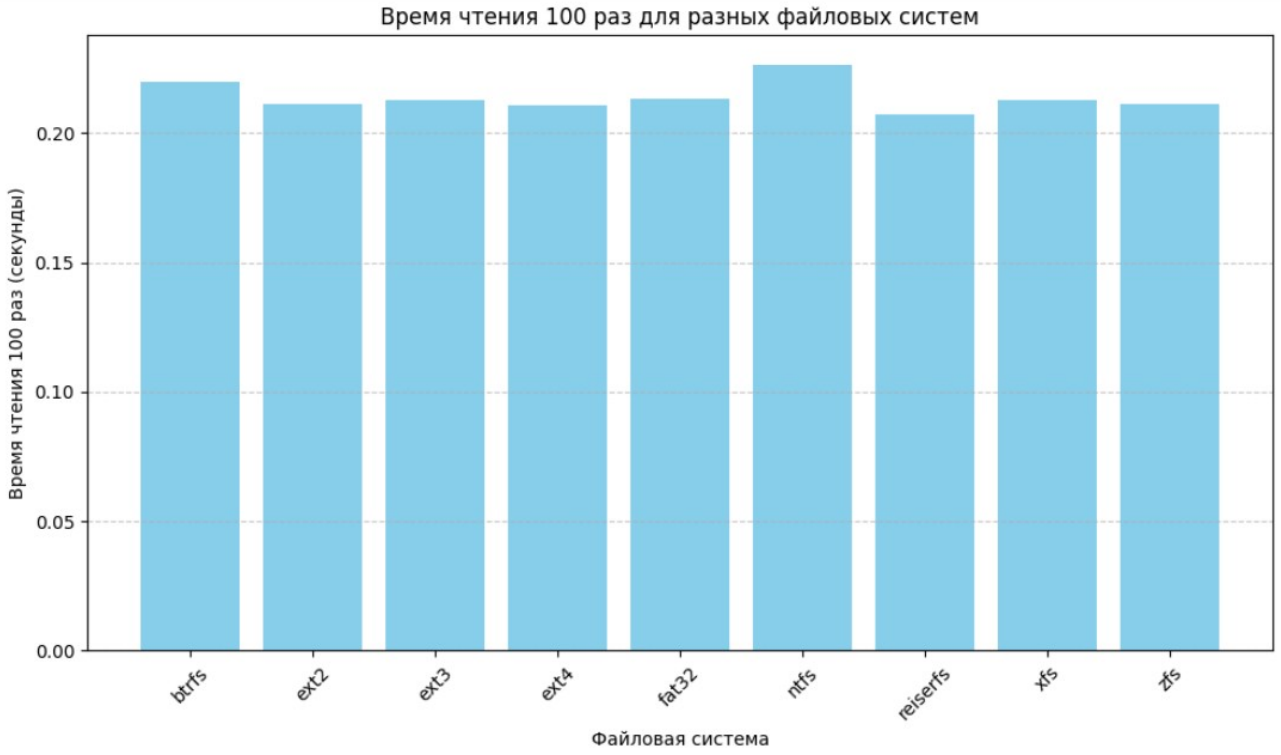
```
#!/bin/bash

RESULTS_FILE="19time_res.log"

mount_points=(
  /mnt/ext2
  /mnt/ext3
  /mnt/ext4
  /mnt/xfs
  /mnt/btrfs
  /mnt/zfs
  /mnt/reiserfs
  /mnt/fat32
  /mnt/ntfs
)
for mp in "${mount_points[@]}"; do
  test_dir="$mp/dir_test"
  echo "Создание каталога dir_test в $mp"
  mkdir -p "$test_dir"
  echo "Создание 1000 подкаталогов в $mp"
  start=$(date +%s.%N)
  for i in $(seq 1 1000); do
    mkdir "$test_dir/dir_$i"
  done
  end=$(date +%s.%N)
  duration=$(echo "$end - $start" | bc)
  result="Шаг19: Время создания 1000 подкаталогов в $mp: $duration секунд"
  echo "$result" | tee -a "$RESULTS_FILE"
done
```

Получил следующий результат

```
Шаг19: Время создания 1000 подкаталогов в /mnt/ext2: 1.900229284 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/ext3: 1.952357547 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/ext4: 2.425826499 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/xfs: 1.970264395 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/btrfs: 1.916782938 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/zfs: 2.110520517 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/reiserfs: 2.000605782 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/fat32: 2.783734632 секунд
Шаг19: Время создания 1000 подкаталогов в /mnt/ntfs: 2.300141789 секунд
~
```

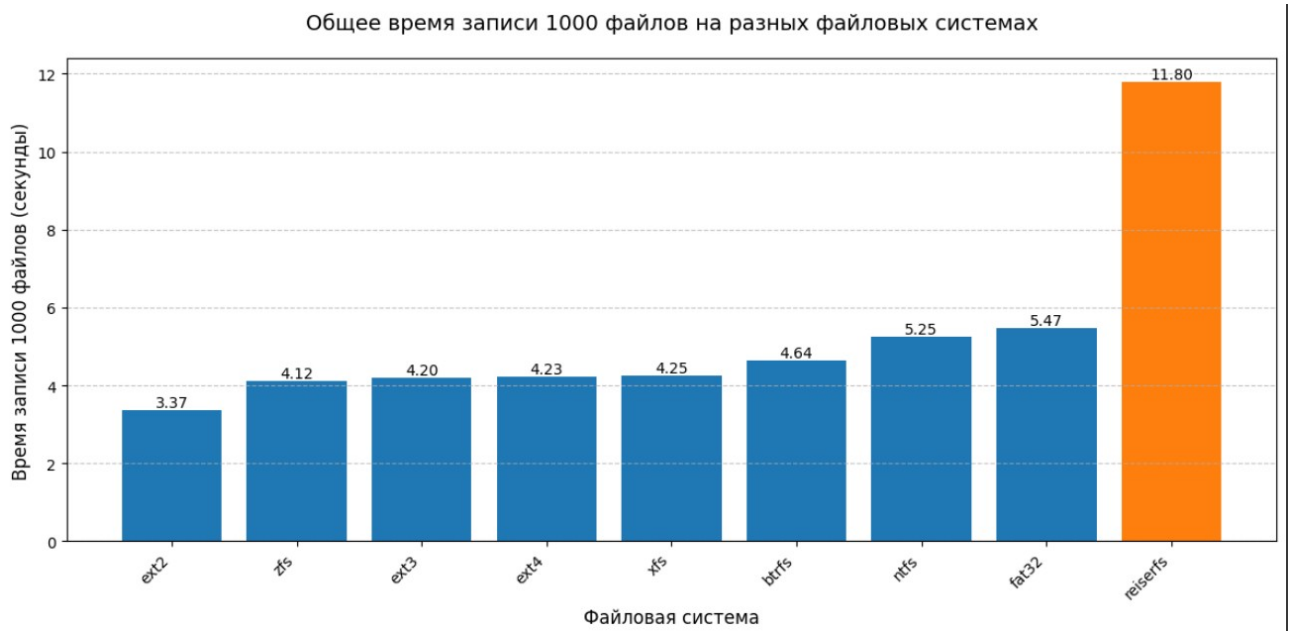
20. Замерить время поиска по созданной структуре для каждой файловой системы — визуализация замеров

Первый тест



Второй тест

Третий тест

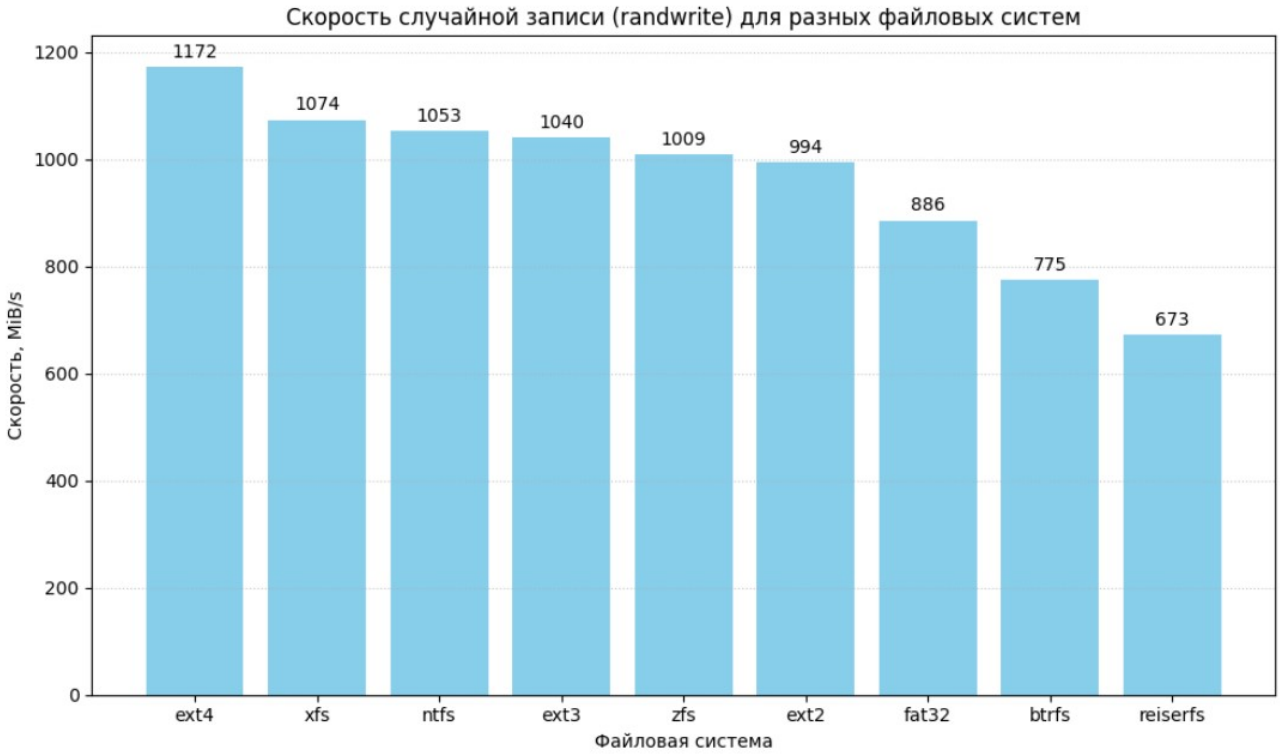


Четвертый тест

Пятый тест


Время поиска по каталогу на разных файловых системах
Время поиска по каталогу (секунды)
0.0
0.1
0.2
0.3
0.4
0.5
0.004
0.028
0.029
0.029
0.030
0.031
0.032
0.032
0.485
ext4
reiserfs
ext3
xfs
zfs
ext2
fat32
btrfs
ntfs
Файловая система

Шестой тест


Скорость случайной записи (randwrite) для разных файловых систем
Скорость, MiB/s
0
200
400
600
800
1000
1200
1172
1074
1053
1040
1009
994
886
775
673
ext4
xfs
ntfs
ext3
zfs
ext2
fat32
btrfs
reiserfs
Файловая система