

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

Факультет безопасности информационных технологий

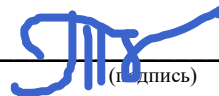
Дисциплина:
«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

«Тестирование файловых систем»

Выполнил:

Рядовой Т.С., студент группы N3252



(подпись)

Проверил:

Чебунин Константин Олегович

(отметка о выполнении)

(подпись)

Санкт-Петербург

2023 г.

СОДЕРЖАНИЕ

Введение.....	4
1 Обычный вариант лабораторной	5
1.1 Задание.....	5
1.2 Ход работы и терминал	5
1.3 Сравнение	7
2 Усиленный вариант лабораторной	8
2.1 Задание.....	8
2.2 Ход работы и терминал	8
2.3 Сравнение	10
Заключение.....	11
Список использованных источников	12

ВВЕДЕНИЕ

Цель работы – выбрать 3 файловые системы, выбрать методику и найти лучшую из них.

В обычном варианте:

- Выбрать 3 файловые системы;
- Выбрать методику и определить лучшую из них.

В усложненном варианте:

- Выбрать экзотические файловые системы;
- Сравнить с системами из обычного варианта лабораторной работы.

1 ОБЫЧНЫЙ ВАРИАНТ ЛАБОРАТОРНОЙ

1.1 Задание

Выбрать и протестировать файловые системы.

1.2 Ход работы и терминал

Для начала создадим 5 файлов для теста 5 файловых систем (ext2, ext4, fat, ntfs, btrfs).

```
root@myarchpc ~# dd if=/dev/zero of=test1.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.54216 s, 412 MB/s
root@myarchpc ~# dd if=/dev/zero of=test2.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.58535 s, 406 MB/s
root@myarchpc ~# dd if=/dev/zero of=test3.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.86685 s, 366 MB/s
root@myarchpc ~# dd if=/dev/zero of=test4.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.9125 s, 360 MB/s
root@myarchpc ~# dd if=/dev/zero of=test5.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.70643 s, 387 MB/s
```

Рисунок 1 – Создание 5 файлов

Далее создаем 5 loop-устройств.

```
root@myarchpc ~# losetup /dev/loop1 test1.bin
root@myarchpc ~# losetup /dev/loop2 test2.bin
root@myarchpc ~# losetup /dev/loop3 test3.bin
root@myarchpc ~# losetup /dev/loop4 test4.bin
root@myarchpc ~# losetup /dev/loop5 test5.bin
```

Рисунок 2 – Создание loop-устройств

Проверим:

```
root@myarchpc ~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop1        7:1    0   1000M  0 loop
loop2        7:2    0   1000M  0 loop
loop3        7:3    0   1000M  0 loop
loop4        7:4    0   1000M  0 loop
loop5        7:5    0   1000M  0 loop
sda          8:0    0  465.8G  0 disk
|-sda1       8:1    0     1G  0 part /boot
`-sda2       8:2    0  464.8G  0 part /home
/
```

Рисунок 3 – Проверка создания loop-устройств

Задаем файловые системы соответственно: ext2, ext4, fat, ntfs, btrfs.

```
root@myarchpc ~# mkfs.ext2 /dev/loop1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 256000 4k blocks and 64000 inodes
Filesystem UUID: 6f976cee-e4f8-4c06-afc5-8282c43fee3c
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@myarchpc ~# mkfs.ext4 /dev/loop2
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 256000 4k blocks and 64000 inodes
Filesystem UUID: 35573f89-989f-4cf2-b2bf-c7d1b2c1f4e7
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Рисунок 4 – Создание файловых систем

```
root@myarchpc ~# mkfs.fat /dev/loop3
mkfs.fat 4.2 (2021-01-31)
root@myarchpc ~# mkfs.ntfs /dev/loop4
Failed to set locale, using default 'C'.
The partition start sector was not specified for /dev/loop4 and it could not be obtained automatically. It has been set to 0.
The number of sectors per track was not specified for /dev/loop4 and it could not be obtained automatically. It has been set to 0.
The number of heads was not specified for /dev/loop4 and it could not be obtained automatically. It has been set to 0.
Cluster size has been automatically set to 4096 bytes.
To boot from a device, Windows needs the 'partition start sector', the 'sectors per track' and the 'number of heads' to be set.
Windows will not be able to boot from this device.
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mknfts completed successfully. Have a nice day.
```

Рисунок 5 – Создание файловых систем

```
root@myarchpc ~# mkfs.btrfs /dev/loop5
btrfs-progs v6.6.1
See https://btrfs.readthedocs.io for more information.

Performing full device TRIM /dev/loop5 (1000.00MiB) ...
NOTE: several default settings have changed in version 5.15, please make sure
this does not affect your deployments:
- DUP for metadata (-m dup)
- enabled no-holes (-O no-holes)
- enabled free-space-tree (-R free-space-tree)

Label: (null)
UUID: a2a0f63c-4faa-4208-a64b-5fc0a1979eab
Node size: 16384
Sector size: 4096
Filesystem size: 1000.00MiB
Block group profiles:
  Data: single 8.00MiB
  Metadata: DUP 50.00MiB
  System: DUP 8.00MiB
SSD detected: yes
Zoned device: no
Incompat features: extref, skinny-metadata, no-holes, free-space-tree
Runtime features: free-space-tree
Checksum: crc32c
Number of devices: 1
Devices:
  ID     SIZE  PATH
  1  1000.00MiB /dev/loop5
```

Рисунок 6 – Создание файловых систем

Теперь проведем тесты с помощью утилиты `hdparm`. Для точности проведем измерение для каждой файловой системы по 3 раза.

```

root@myarchpc ➤ hdparm -tT /dev/loop1

/dev/loop1:
Timing cached reads:   3588 MB in  2.00 seconds = 1796.22 MB/sec
Timing buffered disk reads: 940 MB in  3.00 seconds = 313.08 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop2

/dev/loop2:
Timing cached reads:   3460 MB in  2.00 seconds = 1731.67 MB/sec
Timing buffered disk reads: 1000 MB in  2.69 seconds = 371.19 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop3

/dev/loop3:
Timing cached reads:   3516 MB in  2.00 seconds = 1759.78 MB/sec
Timing buffered disk reads: 992 MB in  3.02 seconds = 328.32 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop4

/dev/loop4:
Timing cached reads:   3490 MB in  2.00 seconds = 1747.03 MB/sec
Timing buffered disk reads: 192 MB in  3.00 seconds =  63.96 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop5

/dev/loop5:
Timing cached reads:   3470 MB in  2.00 seconds = 1736.47 MB/sec
Timing buffered disk reads: 1000 MB in  2.73 seconds = 366.25 MB/sec

```

Рисунок 7 – Первый тест

1.3 Сравнение

Таблица 1 – Результаты теста

	ext2		ext4		fat		ntfs		ntfs	
	cached	buffered	cached	buffered	cached	buffered	cached	buffered	cached	buffered
Тест 1	1796	313	1731	371	1759	328	1747	64	1736	366
Тест 2	1795	325	1926	298	1656	314	1749	78	1736	322
Тест 3	1753	328	1741	323	1763	323	1450	72	1709	323
Среднее	1781,333	322	1799,333	330,6667	1726	321,6667	1648,667	71,33333	1727	337

Вывод: в среднем, лучшим является ext4, худшим – ntfs.

2 УСИЛЕННЫЙ ВАРИАНТ ЛАБОРАТОРНОЙ

2.1 Задание

Проведем аналогичные тесты для «экзотических» файловых систем. Я выбрал следующие: udf, nilfs2, f2fs.

2.2 Ход работы и терминал

Для начала создадим 3 файла для теста 3 файловых систем (udf, nilfs2, f2fs) и создание loop устройств.

```
root@myarchpc ~# dd if=/dev/zero of=test6.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.66514 s, 393 MB/s
root@myarchpc ~# dd if=/dev/zero of=test7.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.88086 s, 364 MB/s
root@myarchpc ~# dd if=/dev/zero of=test8.bin bs=1M count=1000
1000+0 records in
1000+0 records out
1048576000 bytes (1.0 GB, 1000 MiB) copied, 2.65459 s, 395 MB/s
root@myarchpc ~# losetup /dev/loop6 test6.bin
root@myarchpc ~# losetup /dev/loop7 test7.bin
root@myarchpc ~# losetup /dev/loop8 test8.bin
```

Рисунок 8 – Создание 3 файлов

Проверка.

```
root@myarchpc ~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop1        7:1    0   1000M  0 loop
loop2        7:2    0   1000M  0 loop
loop3        7:3    0   1000M  0 loop
loop4        7:4    0   1000M  0 loop
loop5        7:5    0   1000M  0 loop
loop6        7:6    0   1000M  0 loop
loop7        7:7    0   1000M  0 loop
loop8        7:8    0   1000M  0 loop
sda          8:0    0  465.8G  0 disk
|-sda1       8:1    0     1G  0 part /boot
`-sda2       8:2    0  464.8G  0 part /home
/
```

Рисунок 9 – Проверка создания

Задаем файловые системы соответственно: udf, nilfs2, f2fs.

```
root@myarchpc ➜ mkfs -t udf test6.bin
filename=test6.bin
label=LinuxUDF
uuid=656cf32943b412b1
blocksize=512
blocks=2048000
udfrev=2.01
start=0, blocks=64, type=ERASE
start=64, blocks=13, type=VRS
start=77, blocks=19, type=ERASE
start=96, blocks=16, type=MVDS
start=112, blocks=16, type=ERASE
start=128, blocks=16, type=LVID
start=144, blocks=112, type=ERASE
start=256, blocks=1, type=ANCHOR
start=257, blocks=2047480, type=PSPACE
start=2047737, blocks=6, type=ERASE
start=2047743, blocks=1, type=ANCHOR
start=2047744, blocks=96, type=ERASE
start=2047840, blocks=16, type=RVDS
start=2047856, blocks=143, type=ERASE
start=2047999, blocks=1, type=ANCHOR
root@myarchpc ➜ mkfs -t nilfs2 test7.bin
mkfs.nilfs2 (nilfs-utils 2.2.9)
Start writing file system initial data to the device
Blocksize:4096 Device:test7.bin Device Size:1048576000
File system initialization succeeded !!
```

Рисунок 10 – Создание файловых систем

```
root@myarchpc ➜ mkfs -t f2fs test8.bin
F2FS-tools: mkfs.f2fs Ver: 1.16.0 (2023-04-11)
Info: Disable heap-based policy
Info: Debug level = 0
Info: Trim is enabled
Info: Segments per section = 1
Info: Sections per zone = 1
Info: sector size = 512
Info: total sectors = 2048000 (1000 MB)
Info: zone aligned segment0 blkaddr: 512
Info: format version with
"Linux version 6.6.1-zen1-1-zen (linux-zen@archlinux) (gcc (GCC) 13.2.1 20230801, GNU ld (GNU Binutils) 2.41.0) #1 ZEN SMP PREEMPT_DYNAMIC Wed, 08 Nov 2023 16:05:16 +0000"
Info: [test8.bin] Discarding device
Info: Overprovision ratio = 5.450%
Info: Overprovision segments = 27 (GC reserved = 25)
Info: format successful
```

Рисунок 11 – Создание файловых систем

Теперь проведем тесты аналогичные тем, что были в обычном варианте лабораторной.


```

root@myarchpc ➤ hdparm -tT /dev/loop6

/dev/loop6:
Timing cached reads:   3520 MB in  2.00 seconds = 1761.59 MB/sec
Timing buffered disk reads: 1000 MB in  2.99 seconds = 334.14 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop7

/dev/loop7:
Timing cached reads:   3560 MB in  2.00 seconds = 1781.30 MB/sec
Timing buffered disk reads: 918 MB in  3.00 seconds = 305.59 MB/sec
root@myarchpc ➤ hdparm -tT /dev/loop8

/dev/loop8:
Timing cached reads:   2988 MB in  2.00 seconds = 1494.70 MB/sec
Timing buffered disk reads: 798 MB in  3.00 seconds = 265.95 MB/sec

```

Рисунок 12 – Первый тест

2.3 Сравнение

Таблица 2 – Сравнение

	udf		nilfs2		f2fs	
	cached	buffered	cached	buffered	cached	buffered
Тест 1	1761	334	1781	305	1494	266
Тест 2	1758	252	1757	292	1733	275
Тест 3	1501	240	1719	218	1700	276
Среднее	1673,333	275,3333	1752,333	271,6667	1642,333	272,3333

Хоть и не с большим отрывом, в среднем оказалась лучшей система nilfs2.

ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы мне удалось достигнуть поставленных целей:

Обычного варианта:

- Протестировать файловые системы;
- Определить какая из них в среднем лучше и хуже остальных.

Усложненного варианта:

- Протестировать 3 «экзотических» файловых систем;
- Сравнить их.

По итогам проделанной работы стало ясно, что критичной разницы между файловыми системами на проверяемом железе нету. Я бы не выбрал ntfs, так как она показала себя хуже всех. Мой выбор для операционной системы linux – ext4.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://goog-perftools.sourceforge.net/doc/tcmalloc.html>
2. <https://www.kernel.org/doc/gorman/html/understand/understand009.html>