

МИНИСТЕРСТВО 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(1)  
Гоголева Виктора Григорьевича

09.03.01 «Направление подготовки»

Симферополь, 2025

## Ход работы

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

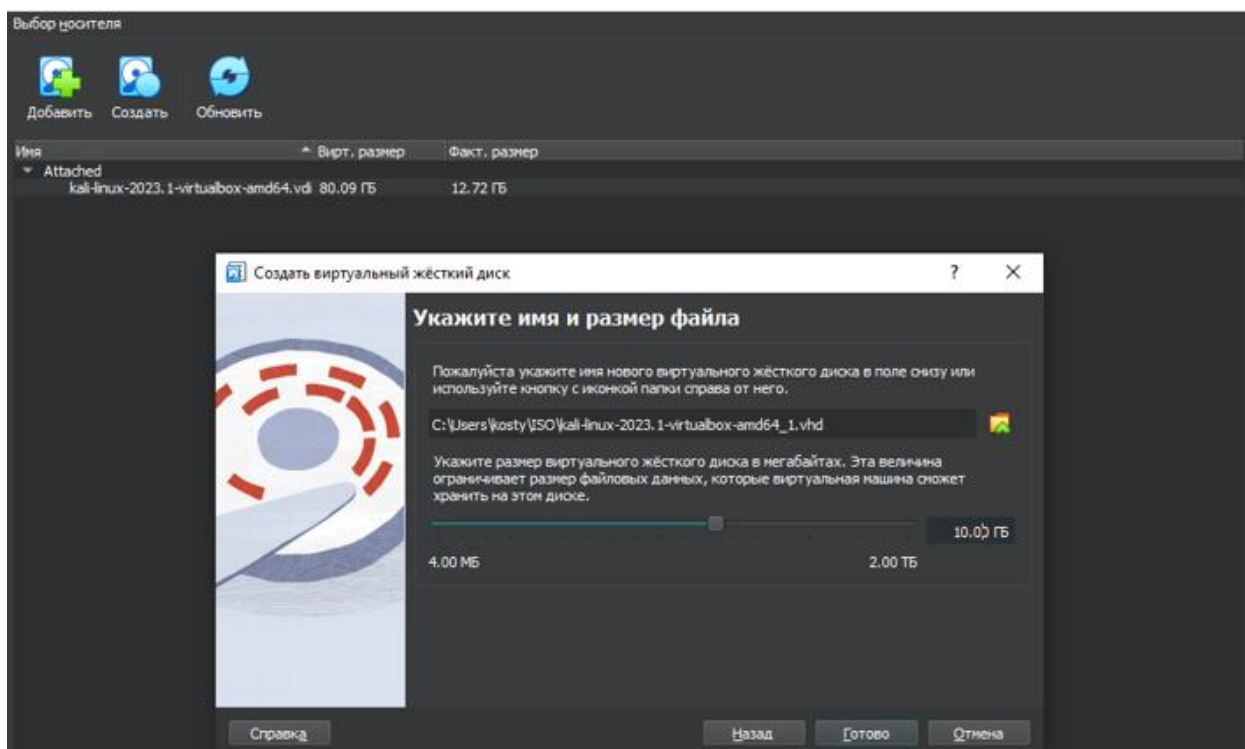


Рисунок – создание виртуального диска

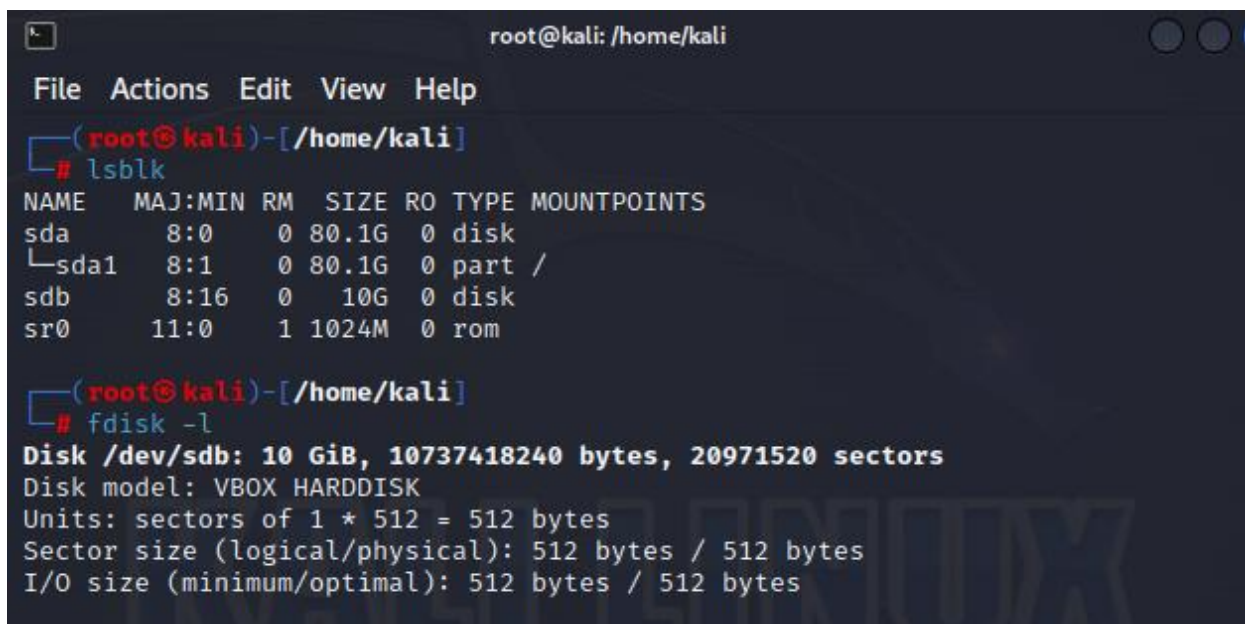


Рисунок – проверка диска и блочных устройств

Как видно в системе появился диск sdb1 на 10Gb, но он пока не размечен и без файловых систем.

```

root@kali: /home/kali

File Actions Edit View Help

Disk: /dev/sdb
Size: 10 GiB, 10737418240 bytes, 20971520 sectors
Label: gpt, identifier: F30F16D8-28D2-C74C-9343-EFD58A6EBAE4

Device          Start      End      Sectors  Size Type
/dev/sdb1        2048      2099199  2097152   1G Linux filesystem
/dev/sdb2       2099200      4196351  2097152   1G Linux filesystem
/dev/sdb3       4196352      6293503  2097152   1G Linux filesystem
/dev/sdb4       6293504      8390655  2097152   1G Linux filesystem
/dev/sdb5       8390656     10487807  2097152   1G Linux filesystem
/dev/sdb6      10487808     12584959  2097152   1G Linux filesystem
/dev/sdb7      12584960     14682111  2097152   1G Linux filesystem
/dev/sdb8      14682112     16779263  2097152   1G Linux filesystem
/dev/sdb9      16779264     18876415  2097152   1G Linux filesystem
>> /dev/sdb10    18876416     20969471  2093056  1022M Linux swap

Partition UUID: CDE593A4-F8F7-4B42-BCDE-0ED8B404A73E
Partition type: Linux swap (0657FD6D-A4AB-43C4-84E5-0933C84B4F4F)

[ Delete ] [ Resize ] [ Quit ] [ Type ] [ Help ] [ Write ]
[ Dump ]

```

Рисунок – разметка созданного диска (я воспользовался утилитой cfdisk)

Создано 10 разделов – каждый размером по 1Gb

```
(root@kali)-[/home/kali]
# fdisk -l
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: F30F16D8-28D2-C74C-9343-EFD58A6EBAE4
```

Device	Start	End	Sectors	Size	Type
/dev/sdb1	2048	2099199	2097152	1G	Linux filesystem
/dev/sdb2	2099200	4196351	2097152	1G	Linux filesystem
/dev/sdb3	4196352	6293503	2097152	1G	Linux filesystem
/dev/sdb4	6293504	8390655	2097152	1G	Linux filesystem
/dev/sdb5	8390656	10487807	2097152	1G	Linux filesystem
/dev/sdb6	10487808	12584959	2097152	1G	Linux filesystem
/dev/sdb7	12584960	14682111	2097152	1G	Linux filesystem
/dev/sdb8	14682112	16779263	2097152	1G	Linux filesystem
/dev/sdb9	16779264	18876415	2097152	1G	Linux filesystem
/dev/sdb10	18876416	20969471	2093056	1022M	Linux swap

```
(root@kali)-[/home/kali]
# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINTS
sda	8:0	0	80.1G	0	disk	
└─sda1	8:1	0	80.1G	0	part	/
sdb	8:16	0	10G	0	disk	
├─sdb1	8:17	0	1G	0	part	
├─sdb2	8:18	0	1G	0	part	
├─sdb3	8:19	0	1G	0	part	
├─sdb4	8:20	0	1G	0	part	
├─sdb5	8:21	0	1G	0	part	
├─sdb6	8:22	0	1G	0	part	
├─sdb7	8:23	0	1G	0	part	
├─sdb8	8:24	0	1G	0	part	
├─sdb9	8:25	0	1G	0	part	
└─sdb10	8:26	0	1022M	0	part	
sr0	11:0	1	1024M	0	rom	

```
(root@kali)-[/home/kali]
#
```

Рисунок – проверка состояние диска и блочных устройств после разметки



## СОЗДАНИЕ РАЗЛИЧНЫХ ФАЙЛОВЫХ СИСТЕМ НА РАЗДЕЛАХ ДИСКА

```

(root@kali)-[/home/kali]
# mkfs.ext2 /dev/sdb1
mke2fs 1.46.6 (1-Feb-2023)
/dev/sdb1 contains a ext2 file system
    created on Mon May 22 09:30:23 2023
Proceed anyway? (y,N) y
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 764de8a8-79ed-4f35-9d9b-fc42d277d195
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

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

"the quieter you become, the more you are able to hear"

(root@kali)-[/home/kali]
# tune2fs -l /dev/sdb1 | grep features
Filesystem features:      ext_attr resize_inode dir_index filetype sparse_sup
er large_file

```

Рисунок – создание файловой системы ext2(Не поддерживает журналирование)

```

(root@kali)-[/home/kali]
# mkfs.ext3 /dev/sdb2 -j
mke2fs 1.46.6 (1-Feb-2023)
/dev/sdb2 contains a ext3 file system
    created on Mon May 22 09:24:22 2023
Proceed anyway? (y,N) y
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 6ccac6d4-7a58-43f8-8d2e-33ad1c8d6f63
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

"the quieter you become, the more you are able to hear"

(root@kali)-[/home/kali]
# tune2fs -l /dev/sdb2 | grep features
Filesystem features:      has_journal ext_attr resize_inode dir_index filetyp
e sparse_super large_file

```

Рисунок – создание файловой системы ext3

```

B Volume
(root@kali)-[/home/kali]
# mkfs.ext4 /dev/sdb3 -j
mke2fs 1.46.6 (1-Feb-2023)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 64aa513a-a423-4897-afdd-e4b6d6019558
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

(root@kali)-[/home/kali]
# tune2fs -l /dev/sdb3 | grep features
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype
e extent 64bit flex_bg sparse_super large_file huge_file dir_nlink extra_isiz
e metadata_csum

```

Рисунок – создание файловой системы ext4(по умолчанию)

```

(root@kali)-[/home/kali]
# mkfs.xfs /dev/sdb4
meta-data=/dev/sdb4          isize=512    agcount=4, agsize=65536 blks
                             sectsz=512    attr=2, projid32bit=1
                             crc=1          finobt=1, sparse=1, rmapbt=0
                             reflink=1       bigtime=1 inobtcount=1 nrext64=0
data      =                  bsize=4096    blocks=262144, 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
                             =             sunit=0 blks, lazy-count=1
realtime  =none             extsz=4096    blocks=0, rtextents=0

```

Рисунок – создание файловой системы xfs

```

(root@kali)-[/home/kali]
# mkfs.btrfs /dev/sdb5 -f
btrfs-progs v6.2
See http://btrfs.wiki.kernel.org for more information.

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:                                302bfffac-4153-4e43-b152-73045f681cc7
Node size:                            16384
Sector size:                           4096
Filesystem size:                       1.00GiB
Block group profiles:
  Data:                                single                8.00MiB
  Metadata:                            DUP                  51.19MiB
  System:                              DUP                   8.00MiB
SSD detected:                          no
Zoned device:                          no
Incompat features: extref, skinny-metadata, no-holes
Runtime features: free-space-tree
Checksum:                               crc32c
Number of devices: 1
Devices:
  ID        SIZE  PATH
  1         1.00GiB /dev/sdb5

(root@kali)-[/home/kali]
# btrfs filesystem show /dev/sdb5
Label: none  uuid: 302bfffac-4153-4e43-b152-73045f681cc7
  Total devices 1 FS bytes used 144.00KiB
    devid    1 size 1.00GiB used 126.38MiB path /dev/sdb5

```

Рисунок – создание файловой системы BTRFS

```

(root@kali)-[/mnt]
# zpool create -f pool1 /dev/sdb6
Defaulting to 4K blocksize (ashift=12) for '/dev/sdb6'
invalid vdev specification
the following errors must be manually repaired:
/dev/sdb6 is part of active pool 'pool1'

(root@kali)-[/mnt]
# zpool scrub pool1

(root@kali)-[/mnt]
# zpool list
NAME      SIZE  ALLOC   FREE      CAP  DEDUP  HEALTH  ALTROOT
pool1    1016M   964K  1015M       0%  1.00x  ONLINE   -

(root@kali)-[/mnt]
# zfs create pool1/files

(root@kali)-[/mnt]
# zfs create pool1/media

(root@kali)-[/mnt]
# zpool list
NAME      SIZE  ALLOC   FREE      CAP  DEDUP  HEALTH  ALTROOT
pool1    1016M  1.22M  1015M       0%  1.00x  ONLINE   -

(root@kali)-[/mnt]
# zfs list
NAME                USED  AVAIL  REFER  MOUNTPOINT
pool1                984K   983M   128K   /pool1
pool1/data            112K   983M   112K   /pool1/data
pool1/files           112K   983M   112K   /pool1/files
pool1/media           112K   983M   112K   /pool1/media
pool1/mediaclear      112K   983M   112K   /pool1/mediaclear

```

Рисунок – создание файловой системы ZFS

```

(root@kali)-[/home/kali/OS]
# mkfs.vfat /dev/sdb8
mkfs.fat 4.2 (2021-01-31)

```

Рисунок – создание файловой системы FAT32



```
(root@kali)-[/home/kali/OS]
# mkfs.ntfs /dev/sdb9
Cluster size has been automatically set to 4096 bytes.
Initializing device with zeroes: 100% - Done.
Creating NTFS volume structures.
mknfts completed successfully. Have a nice day.
```

Рисунок – создание файловой системы ntfs

Создать раздел swar и активировать его в системе. Создать файловую систему типа swar в файле и активировать в системе.

```
(root@kali)-[/home/kali/OS]
# mkswap /dev/sdb10
Setting up swapspace version 1, size = 1022 MiB (1071640576 bytes)
no label, UUID=e0e69278-c129-4d9c-93f7-eee278880c90

(root@kali)-[/home/kali/OS]
# swapon -s
```

Filename	Type	Size	Used	Priority
/swapfile	file	1048572	247580	-2

```
(root@kali)-[/home/kali/OS]
# swapon -U e0e69278-c129-4d9c-93f7-eee278880c90

(root@kali)-[/home/kali/OS]
# swapon -s
```

Filename	Type	Size	Used	Priority
/swapfile	file	1048572	247324	-2
/dev/sdb10	partition	1046524	0	-3

Рисунок – создание swar- раздела

## АВТОМАТИЗАЦИЯ МОНТИРОВАНИЯ

Настроить ОС, чтобы все файловые системы монтировались при старте системы.

**устройство** **точка\_монтирования** **файловая\_система** **опции** **резерв{0,1}** **проверка{0,1,2}**

Теперь подробнее рассмотрим что означает каждый пункт:

- **Устройство** - это раздел диска, который вам нужно примонтировать. Его можно указать в формате файла устройства Linux в каталоге /dev/, например, /dev/sda1 или с помощью уникального идентификатора UUID, тогда формат записи будет таким UUID="XXXX-XXXX-XXXX-XXXX", также возможна запись с помощью метки, например, LABEL=home;
- **Файловая система** указывает в какой файловой системе нужно монтировать это устройство, например, ext4, ext3, ext2, btrfs;
- **Точка монтирования** - куда нужно примонтировать это устройство, например, /home, /boot, /mnt;
- **Опции** - параметры монтирования файловой системы, рассмотрим подробнее ниже;
- **Проверка** - указывает в какой очереди нужно проверять устройство на ошибки, 1 - в первую очередь, 2 - вторую, 0 - не проверять;
- **Резерв** - указывает нужно ли делать резервную копию раздела, может принимать значения только 0 и 1.

Рисунок – синтаксис команды mount

```
(root@kali)-[/mnt]
# mount /dev/sdb1 /mnt/ext2

(root@kali)-[/mnt]
# mount /dev/sdb2 /mnt/ext3

(root@kali)-[/mnt]
# mount /dev/sdb3 /mnt/ext4

(root@kali)-[/mnt]
# mount /dev/sdb4 /mnt/xfs

(root@kali)-[/mnt]
# mount /dev/sdb5 /mnt/btrfs

(root@kali)-[/mnt]
# mount /dev/sdb6 /mnt/zfs
mount: /mnt/zfs: unknown filesystem type 'zfs_member'.
dmesg(1) may have more information after failed mount system call.

(root@kali)-[/mnt]
# mount /dev/sdb7 /mnt/reiserfs

(root@kali)-[/mnt]
# mount /dev/sdb8 /mnt/fat32

(root@kali)-[/mnt]
# mount /dev/sdb9 /mnt/ntfs
```

Рисунок - Монтирование форматированных разделов в директории системы

```
(root@kali)-[/mnt]
# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0 80.1G  0 disk
└─sda1       8:1    0 80.1G  0 part /
sdb          8:16   0   10G  0 disk
├─sdb1       8:17   0    1G  0 part /mnt/ext2
├─sdb2       8:18   0    1G  0 part /mnt/ext3
├─sdb3       8:19   0    1G  0 part /mnt/ext4
├─sdb4       8:20   0    1G  0 part /mnt/xfs
├─sdb5       8:21   0    1G  0 part /mnt/btrfs
├─sdb6       8:22   0    1G  0 part
├─sdb7       8:23   0    1G  0 part /mnt/reiserfs
├─sdb8       8:24   0    1G  0 part /mnt/fat32
├─sdb9       8:25   0    1G  0 part /mnt/ntfs
└─sdb10      8:26   0 1022M  0 part [SWAP]
sr0         11:0    1 1024M  0 rom
```

Рисунок – список блочных устройств и их точек монтирования

```
File  Машина  Вид  Ввод  Устройства  Справка
Applications
File Actions Edit View Help
GNU nano 7.2
UUID=1842db13-cb07-4ed0-9bd6-56e19c5665a2 / ext4 defaults,errors
/swapfile none swap defaults 0 0
/dev/sdb1 /mnt/ext2 ext2 defaults 0 2
/dev/sdb2 /mnt/ext3 ext3 defaults 0 2
/dev/sdb3 /mnt/ext4 ext4 defaults 0 2
/dev/sdb4 /mnt/xfs xfs defaults 0 2
/dev/sdb5 /mnt/btrfs btrfs defaults 0 2
/dev/sdb7 /mnt/reiserfs reiserfs defaults 0 2
/dev/sdb8 /mnt/fat32 vfat defaults 0 2
/dev/sdb9 /mnt/ntfs ntfs defaults 0 2
/dev/sdb10 swap swap defaults 0 0
```

Рисунок – вывод всех перемонтированных разделов и ФС на них после перезагрузки

## СКРИПТЫ

1. 15 Протестировать все разделы для операций чтения маленьких файлов (16кб), цикл не менее 100 раз



```

root@kali: /home/kali/OS/Laba_7
File Actions Edit View Help
GNU nano 7.2 task_15.sh
#!/bin/bash

fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

file_size="16k"
count_reads=100

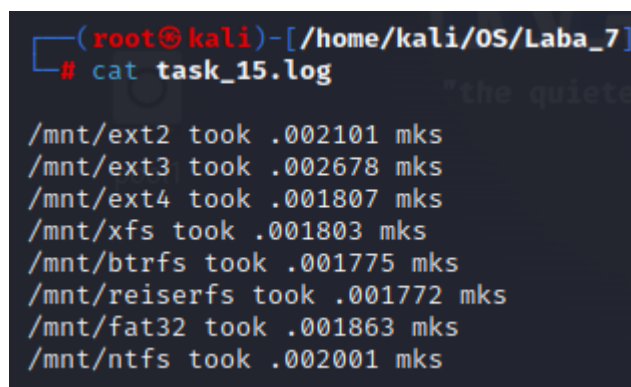
echo "" > task_15.log

for partition in "${fail_system[@]}; do
    dd if=/dev/zero of="$partition/testfile" bs="$file_size" count=1
    start_time=$(date +%s.%6N)
    dd if="$partition/testfile" of=/dev/null bs="$file_size" count=100
    end_time=$(date +%s.%6N)

    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_15.log

    rm "$partition/testfile"
done
  
```

Рисунок – скрипт тестирования разделов на чтение маленьких файлов



```

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_15.log

/mnt/ext2 took .002101 mks
/mnt/ext3 took .002678 mks
/mnt/ext4 took .001807 mks
/mnt/xfs took .001803 mks
/mnt/btrfs took .001775 mks
/mnt/reiserfs took .001772 mks
/mnt/fat32 took .001863 mks
/mnt/ntfs took .002001 mks
  
```

Рисунок – вывод лога по чтению маленьких файлов

16. Протестировать все файловые системы для операций чтения больших файлов(500мб)

```

GNU nano 7.2 task_16.sh
#!/bin/bash

fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

file_size="500M"

echo "" > task_16.log

for partition in "${fail_system[@]}; do
    dd if=/dev/zero of="$partition/testfile" bs="$file_size" count=1

    start_time=$(date +%s.%6N)
    dd if="$partition/testfile" of=/dev/null bs="$file_size" count=100
    end_time=$(date +%s.%6N)

    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_16.log

    rm "$partition/testfile"
done

```

Рисунок – скрипт тестирования больших файлов

```

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_16.log

/mnt/ext2 took .180637 mks
/mnt/ext3 took .179480 mks
/mnt/ext4 took .140054 mks
/mnt/xfs took .175682 mks
/mnt/btrfs took .219337 mks
/mnt/reiserfs took .219066 mks
/mnt/fat32 took .185229 mks
/mnt/ntfs took .319330 mks

```

Рисунок – вывод результатов тестирования чтения больших файлов

## 17.Протестировать операции записи для маленьких файлов



```

GNU nano 7.2                                     root@kali: /home/kali/OS/Laba_7
#!/bin/bash                                     task_17.sh *

fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

file_size="16k"

echo "" > task_17.log

for partition in "${fail_system[@]}; do
    start_time=$(date +%s.%6N)
    dd if=/dev/zero of="$partition/testfile" bs="$file_size" count=100
    end_time=$(date +%s.%6N)
    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_17.log
    rm "$partition/testfile"
done

```

Рисунок – скрипт тестирования операций записи для маленьких файлов

```

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_17.log

/mnt/ext2 took .004187 mks
/mnt/ext3 took .004632 mks
/mnt/ext4 took .003327 mks
/mnt/xfs took .002662 mks
/mnt/btrfs took .002889 mks
/mnt/reiserfs took .003964 mks
/mnt/fat32 took .005763 mks
/mnt/ntfs took .025588 mks

```

Рисунок – результат тестирования операций чтения для маленьких файлов

## 18. Протестировать операции записи для больших файлов



```

GNU nano 7.2 task_18.sh *
fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

file_size="500M"
echo "" > task_18.log

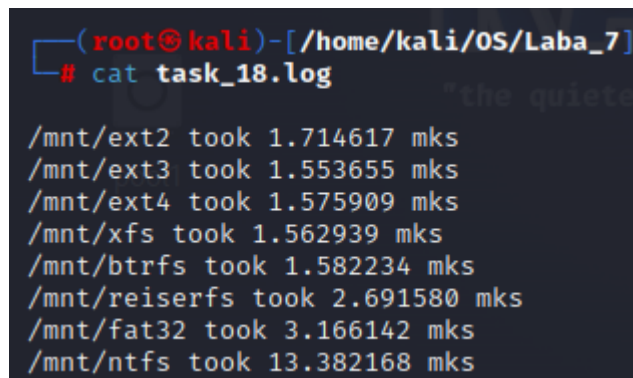
for partition in "${fail_system[@]}; do

    start_time=$(date +%s.%6N)
    dd if=/dev/zero of="$partition/testfile" bs="$file_size" count=100
    end_time=$(date +%s.%6N)

    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_18.log

    rm "$partition/testfile"
done
  
```

Рисунок – скрипт тестирования операций записи для больших файлов



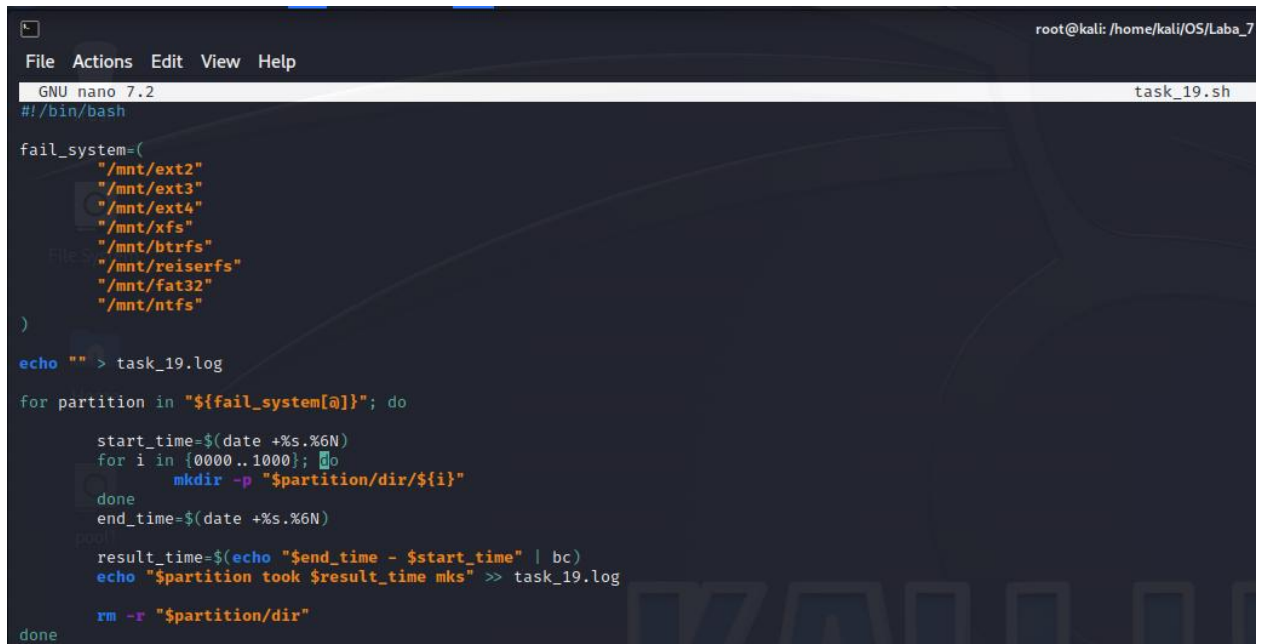
```

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_18.log

/mnt/ext2 took 1.714617 mks
/mnt/ext3 took 1.553655 mks
/mnt/ext4 took 1.575909 mks
/mnt/xfs took 1.562939 mks
/mnt/btrfs took 1.582234 mks
/mnt/reiserfs took 2.691580 mks
/mnt/fat32 took 3.166142 mks
/mnt/ntfs took 13.382168 mks
  
```

Рисунок – вывод результатов тестирования операций записи больших файлов

19. Создать структуру каталогов с не менее чем 1000 подкаталогов в каждой файловой системе, замерить время создания.



```

root@kali: /home/kali/OS/Laba_7
File Actions Edit View Help
GNU nano 7.2 task_19.sh
#!/bin/bash

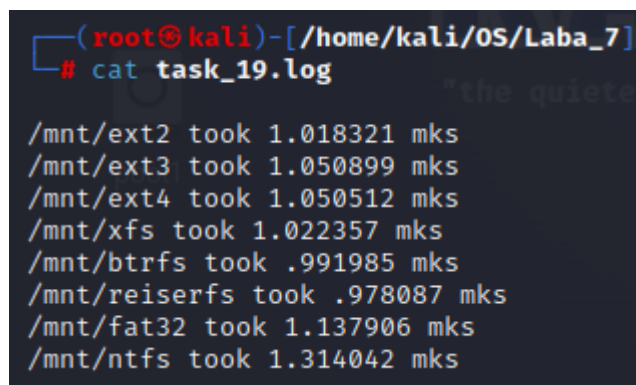
fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

echo "" > task_19.log

for partition in "${fail_system[@]}; do

    start_time=$(date +%s.%6N)
    for i in {0000..1000}; do
        mkdir -p "$partition/dir/${i}"
    done
    end_time=$(date +%s.%6N)
    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_19.log
    rm -r "$partition/dir"
done
  
```

Рисунок – листинг скрипта для создания структуры каталогов



```

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_19.log

/mnt/ext2 took 1.018321 mks
/mnt/ext3 took 1.050899 mks
/mnt/ext4 took 1.050512 mks
/mnt/xfs took 1.022357 mks
/mnt/btrfs took .991985 mks
/mnt/reiserfs took .978087 mks
/mnt/fat32 took 1.137906 mks
/mnt/ntfs took 1.314042 mks
  
```

Рисунок – вывод информации о времени создания для каждой файловой системы

## 20. Замерить время поиска по созданной структуре для каждой файловой системы

```

root@kali: /home/kali/OS/Laba_7
GNU nano 7.2 task_20.sh
#!/bin/bash

fail_system=(
    "/mnt/ext2"
    "/mnt/ext3"
    "/mnt/ext4"
    "/mnt/xfs"
    "/mnt/btrfs"
    "/mnt/reiserfs"
    "/mnt/fat32"
    "/mnt/ntfs"
)

echo "" > task_20.log

for partition in "${fail_system[@]}; do

    for i in {0000..1000}; do
        mkdir -p "$partition/dirs/dir${i}"
    done

    start_time=$(date +%s.%6N)
    find "$partition/dirs" -type d -name "dir1000"
    end_time=$(date +%s.%6N)

    result_time=$(echo "$end_time - $start_time" | bc)
    echo "$partition took $result_time mks" >> task_20.log

    rm -r "$partition/dirs"
done

```

Рисунок – листинг скрипта для вычисления времени поиска по созданной структуре для каждой ФС

```

(root@kali)-[/home/kali/OS/Laba_7]
# ./task_20.sh
/mnt/ext2/dirs/dir1000
/mnt/ext3/dirs/dir1000
/mnt/ext4/dirs/dir1000
/mnt/xfs/dirs/dir1000
/mnt/btrfs/dirs/dir1000
/mnt/reiserfs/dirs/dir1000
/mnt/fat32/dirs/dir1000
/mnt/ntfs/dirs/dir1000

(root@kali)-[/home/kali/OS/Laba_7]
# cat task_20.log

/mnt/ext2 took .011792 mks
/mnt/ext3 took .012039 mks
/mnt/ext4 took .012764 mks
/mnt/xfs took .011751 mks
/mnt/btrfs took .012871 mks
/mnt/reiserfs took .010239 mks
/mnt/fat32 took .012908 mks
/mnt/ntfs took .236815 mks

```

Рисунок - вывод результатов замеров времени поиска по созданной структуре для каждой ФС

## ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ РАБОТЫ

21. Представить результаты каждого эксперимента по замеру производительности операции в файловой системе в виде графика

	0	1	2	3	4	5	6	7
	ext2	ext3	ext4	xfs	btrfs	reiserfs	fat32	ntfs
1	0,0021	0,0027	0,001807	0,001803	0,001775	0,001772	0,001863	0,002001
2	0,18	0,179	0,14	0,17	0,22	0,219	0,185	0,32
3	0,0042	0,0046	0,0033	0,0027	0,0029	0,0039	0,0057	0,025
4	1,71	1,55	1,57	1,56	1,58	2,69	3,16	13,38
5	1,01	1,05	1,05	1,02	0,99	0,97	1,13	1,31
6	0,011	0,012	0,013	0,012	0,013	0,01	0,013	0,23

Рисунок – занес результаты вычислений по каждой из ФС в таблицу Excel

