

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №2**  
**по дисциплине «Операционные системы»**  
**Тема: Файловые системы Unix-подобных ОС**

Студентка гр. 1304

Чернякова В.А.

Преподаватель

Душутина Е.В.

Санкт-Петербург

2023

### Цель работы.

Проанализировать функциональное назначение структурных элементов дерева ФС. Определить размещение корневого каталога (корневой ФС).

### Выполнение работы.

1. Ознакомимся с типами файлов Linux.

Можно выделить следующие:

Типы файлов		Назначение
Обычные файлы	—	Хранение символьных и двоичных данных
Каталоги	d	Организация доступа к файлам
Символьные ссылки	l	Предоставление доступа к файлам, расположенных на любых носителях
Блочные устройства	b	Предоставление интерфейса для взаимодействия с аппаратным обеспечением компьютера
Символьные устройства	c	
Каналы	p	Организация взаимодействия процессов в операционной системе
Сокеты	s	

Применяя утилиту ls отфильтруем по одному примеру каждого типа файла используемой ФС.

Команда для такой фильтрации выглядит следующим образом:

```
ls -Rl | grep ^(-, d, l, b, c, p, s)
```

Ключ `-l` позволяет выводить подробный список, `-R` отвечает за рекурсивное отображение поддиректорий.

Команда `grep` позволяет осуществить поиск нужного типа файла, для этого после `^` необходимо указать соответствующий символ.

- Обычные файлы

```
lera2003@Valeriya:~$ ls /bin -Rl | grep ^-
-rwxr-xr-x 1 root root 1037528 июл 12 2019 bash
-rwxr-xr-x 1 root root 31352 июл 4 2019 bunzip2
-rwxr-xr-x 1 root root 1984584 мар 7 2019 busybox
-rwxr-xr-x 1 root root 31352 июл 4 2019 bzip2
-rwxr-xr-x 1 root root 2140 июл 4 2019 bzip2diff
-rwxr-xr-x 1 root root 4877 июл 4 2019 bzip2grep
-rwxr-xr-x 1 root root 3642 июл 4 2019 bzip2recover
-rwxr-xr-x 1 root root 14672 июл 4 2019 bzip2recover
-rwxr-xr-x 1 root root 1297 июл 4 2019 bzip2more
-rwxr-xr-x 1 root root 52080 мар 2 2017 cat
-rwxr-xr-x 1 root root 14752 фев 7 2016 chacl
```

- Каталоги

```
lera2003@Valeriya:~$ ls -Rl | grep ^d
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Desktop
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Documents
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Downloads
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Music
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Pictures
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Public
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Templates
drwxr-xr-x 2 lera2003 lera2003 4096 мар 14 01:58 Videos
```

- Символьные ссылки

```
lera2003@Valeriya:~$ ls /bin -Rl | grep ^l
lrwxrwxrwx 1 root root 6 мар 14 01:49 bzip2diff -> bzip2diff
lrwxrwxrwx 1 root root 6 мар 14 01:49 bzip2grep -> bzip2grep
lrwxrwxrwx 1 root root 6 мар 14 01:49 bzip2recover -> bzip2recover
lrwxrwxrwx 1 root root 6 мар 14 01:49 bzip2more -> bzip2more
lrwxrwxrwx 1 root root 8 мар 14 01:49 dnsdomainname -> hostname
lrwxrwxrwx 1 root root 8 мар 14 01:49 domainname -> hostname
lrwxrwxrwx 1 root root 8 мар 14 01:49 lessfile -> lesspipe
lrwxrwxrwx 1 root root 4 мар 14 01:49 lsmod -> kmod
```

- Блочные устройства

```
lera2003@Valeriya:~$ ls /dev -Rl | grep ^b
brw-rw---- 1 root disk 7, 0 мар 16 01:05 loop0
brw-rw---- 1 root disk 7, 1 мар 15 01:43 loop1
brw-rw---- 1 root disk 7, 2 мар 15 01:43 loop2
brw-rw---- 1 root disk 7, 3 мар 15 01:43 loop3
brw-rw---- 1 root disk 7, 4 мар 15 01:43 loop4
brw-rw---- 1 root disk 7, 5 мар 15 01:43 loop5
brw-rw---- 1 root disk 7, 6 мар 15 01:43 loop6
brw-rw---- 1 root disk 7, 7 мар 15 01:43 loop7
```

- Символьные устройства

```
lera2003@Valeriya:~$ ls /dev -Rl | grep ^c
crw-r--r-- 1 root root 10, 235 map 15 01:43 autofs
crw-rw---- 1 root disk 10, 234 map 16 01:10 btrfs-control
crw----- 1 root root 5, 1 map 15 01:43 console
crw----- 1 root root 10, 60 map 15 01:43 cpu_dma_latency
crw----- 1 root root 10, 203 map 15 01:43 cuse
crw----- 1 root root 10, 62 map 15 01:43 ecryptfs
crw-rw---- 1 root video 29, 0 map 15 01:43 fb0
crw-rw-rw- 1 root root 1, 7 map 15 01:43 full
crw-rw-rw- 1 root root 10, 229 map 15 01:43 fuse
crw----- 1 root root 244, 0 map 15 02:26 hidraw0
```

- Каналы

```
prw----- 1 root root 0 map 15 01:43 1.ref
prw----- 1 root root 0 map 15 01:43 2.ref
prw----- 1 root root 0 map 15 01:44 6.ref
prw----- 1 root root 0 map 15 01:44 7.ref
prw----- 1 root root 0 map 15 01:44 8.ref
prw----- 1 root root 0 map 15 01:55 9.ref
prw----- 1 root root 0 map 15 01:43 fifo
```

- Сокеты

```
srw-rw-rw- 1 root root 0 map 15 01:43 acpid.socket
srw-rw-rw- 1 root root 0 map 16 01:05 snapd-snap.socket
srw-rw-rw- 1 root root 0 map 16 01:05 snapd.socket
srw-rw-rw- 1 root root 0 map 15 01:43 socket
srw-rw-rw- 1 root root 0 map 15 01:43 cups.sock
srw-rw-rw- 1 root root 0 map 15 01:43 system_bus_socket
srwxr-xr-x 1 root root 0 map 15 01:43 private-dhcp
srwx----- 1 root root 0 map 15 01:43 cgroups-agent
srw----- 1 root root 0 map 15 01:43 fsck.progress
srwxrwxrwx 1 root root 0 map 15 01:43 notify
srwxrwxrwx 1 root root 0 map 15 01:43 private
```

Теперь создадим скрипт, который будет выводить все типы файлов в директории, переданной в качестве аргумента.

```
#!/bin/bash

#переменная, отвечающая за типы файлов
f_types='- d l b c p s'

#цикл поиска файлов для каждого типа
for i in $f_types
do
    #выводим текущий ключ - какой тип файла найден или нет
    echo $i
    #осуществляем поиск файла с таким типом
    find_file=`ls -lR $1 | grep ^$i | head -1`
    #проверяем, не пуста ли переменная, то есть найден ли файл
    if [[ -n $find_file ]]
    then
        #запоминаем нужную информацию про файл
        inf=`ls -lR $1 | grep ^$i | head -1 | cut -b 53-1000`
        #выводим полный путь к файлу и информацию о нем
        echo "$find_file - `pwd`/$inf"
    else
        echo "file not found!"
    fi
done
```

Результат работы скрипта.

```
lera2003@Valeriya:~$ ./scripts/script1.sh /dev
-
-rwx----- 1 lera2003 lera2003 67108904 map 16 01:46 pulse-shm-1168084920 - /home/lera2003/:46 pulse-shm-1168084920
d
drwxr-xr-x 2 root root          300 map 16 01:44 block - /home/lera2003/block
l
lrwxrwxrwx 1 root root           3 map 16 01:47 cdrom -> sr0 - /home/lera2003/cdrom -> sr0
b
brw-rw---- 1 root disk          7,  0 map 16 01:47 loop0 - /home/lera2003/loop0
c
crw-r--r-- 1 root root        10, 235 map 16 01:47 autofs - /home/lera2003/autofs
p
file not found!
s
file not found!
lera2003@Valeriya:~$ ./scripts/script1.sh /bin
-
-rwxr-xr-x 1 root root 1037528 июл 12  2019 bash - /home/lera2003/
d
file not found!
l
lrwxrwxrwx 1 root root           6 map 14 01:49 bzcmp -> bzdiff - /home/lera2003/ -> bzdiff
b
file not found!
c
file not found!
p
file not found!
s
file not found!
```

2. Напишем скрипт, который находит все жесткий ссылки на заданный файл.

```
#!/bin/sh
#проверяем, заданы ли входные параметры
if [ $# -lt 1 ]
then
    #выводим ошибку - входные параметры не заданы
    echo "$0: Error: No file"
else
    #запоминаем имя файла
    filename=$1
    #получение жесткой ссылки на файл
    inode=`ls -li $filename | cut -c 1-8`
    #поиск жесткой ссылки рекурсивно по каталогу
    ls -lRi | grep "^$inode" > hardlinks.txt
fi
```

Результат работы скрипта.

```
lera2003@Valeriya:~$ ./scripts/script2.sh ./Documents/laba2.txt
948804 -rw-rw-r-- 2 lera2003 lera2003 167 map 16 02:45 laba2.txt
948804 -rw-rw-r-- 2 lera2003 lera2003 167 map 16 02:45 lb2_hard
link
```

3. Для файла laba2.txt сформируем символьные ссылки различными способами.

- ln -s

```
lera2003@Valeriya:~/Documents$ ln -s laba2.txt lab2_1.txt
lera2003@Valeriya:~/Documents$ ls -l lab2_1.txt
lrwxrwxrwx 1 lera2003 lera2003 9 map 16 03:00 lab2_1.txt -> lab2.txt
```

- cp -s

```
lera2003@Valeriya:~/Documents$ cp -s lab2_1.txt lab2_3.txt
lera2003@Valeriya:~/Documents$ ls -l lab2_3.txt
lrwxrwxrwx 1 lera2003 lera2003 9 map 16 03:04 lab2_3.txt -> lab2.txt
```

Создадим скрипт, перечисляющий все полно именные символьные ссылки на файл, размещаемые в разных местах файлового дерева.

```
lera2003@Valeriya: ~  
Файл Правка Вид Поиск Терминал Справка  
#!/bin/sh  
  
#проверка на вводимось параметров  
if [ $# -lt 1 ]  
then  
    #вывод ошибки  
    echo $0: Error: no file  
else  
    #поиск символьных ссылок в каталоге рекурсивно и их запись в файл  
    ls -lRna / | grep "$1 ->" | grep ^l > file_links.txt  
  
    #подсчет количества символьных ссылок  
    cat file_links.txt  
fi
```

Результат работы скрипта.

```
lera2003@Valeriya:~/Documents$ cat file_links.txt  
lrwxrwxrwx 1 lera2003 lera2003 9 мар 16 03:00 laba2_1.txt -> laba2.txt  
lrwxrwxrwx 1 lera2003 lera2003 9 мар 16 03:04 laba2_3.txt -> laba2.txt  
total 2
```

Найдем все символьные ссылки на cc.

```
lera2003@Valeriya:~$ cat file_links.txt  
lrwxrwxrwx 1 0 0 12 мар 14 01:49 cc -> /usr/bin/gcc  
lrwxrwxrwx 1 0 0 3 фев 2 2020 cc -> gcc  
lrwxrwxrwx 1 0 0 20 мар 14 01:50 cc -> /etc/alternatives/cc  
lrwxrwxrwx 1 0 0 5 мар 14 01:50 gcc -> gcc-5  
lrwxrwxrwx 1 0 0 9 мар 14 01:50 gcc -> qtchooser  
lrwxrwxrwx 1 0 0 5 мар 14 01:50 x86_64-linux-gnu-gcc -> gcc-5  
lrwxrwxrwx 1 0 0 3 мар 14 01:52 cc -> gcc  
lrwxrwxrwx 1 0 0 3 мар 14 01:52 gcc -> cpp  
lrwxrwxrwx 1 0 0 12 фев 19 2016 d430-dg-ccc -> d430c-dg-ccc  
lrwxrwxrwx 1 0 0 17 фев 19 2016 d430-unix-25-ccc -> d430c-unix-25-ccc  
lrwxrwxrwx 1 0 0 14 фев 19 2016 d430-unix-ccc -> d430c-unix-ccc  
lrwxrwxrwx 1 0 0 16 фев 19 2016 d430-unix-s-ccc -> d430c-unix-s-ccc  
lrwxrwxrwx 1 0 0 17 фев 19 2016 d430-unix-sr-ccc -> d430c-unix-sr-ccc  
lrwxrwxrwx 1 0 0 16 фев 19 2016 d430-unix-w-ccc -> d430c-unix-w-ccc  
lrwxrwxrwx 1 0 0 58 мар 14 01:53 qconf.cc -> ../../../../linux-headers-4.15.0-112/scripts/kconf  
ig/qconf.cc  
lrwxrwxrwx 1 0 0 58 апр 13 2021 qconf.cc -> ../../../../linux-headers-4.15.0-142/scripts/kconf  
ig/qconf.cc
```

4. Найдем все символьные ссылки на файл.

```
lera2003@Valeriya:~$ ls -lRa | grep laba2.txt | grep ^l  
lrwxrwxrwx 1 lera2003 lera2003 9 мар 16 03:00 laba2_1.txt -> laba2.txt  
lrwxrwxrwx 1 lera2003 lera2003 9 мар 16 03:04 laba2_3.txt -> laba2.txt
```

5. Изучим утилиту find.

Команда `find` - это одна из наиболее важных и часто используемых утилит системы Linux. Это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах,



например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

- Поиск файла по имени.

```
lera2003@Valeriya:~$ find . -name "file_links.txt"
./file_links.txt
./Documents/file_links.txt
```

- Поиск по типу файла.

```
lera2003@Valeriya:~$ find /etc -type l
/etc/dictionaries-common/words
/etc/rc4.d/S04cups
/etc/rc4.d/S04cups-browsed
/etc/rc4.d/S02irqbalance
/etc/rc4.d/S03bluetooth
/etc/rc4.d/S05ondemand
/etc/rc4.d/S03avahi-daemon
/etc/rc4.d/S05plymouth
/etc/rc4.d/S02speech-dispatcher
/etc/rc4.d/S05grub-common
/etc/rc4.d/S01rsyslog
```

- Поиск файлов по размеру.

```
lera2003@Valeriya:~$ find ./ -size +1M -size -25M
./snap/notepad-plus-plus/common/.wine/drive_c/Program Files (x86)/Windows NT/Accessories/wordpad.exe
./snap/notepad-plus-plus/common/.wine/drive_c/Program Files (x86)/Common Files/System/OLE DB/msdaps.dll
./snap/notepad-plus-plus/common/.wine/drive_c/Program Files (x86)/Common Files/System/OLE DB/oledb32.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/twain_64/gphoto2.ds
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/wbem/wbemprox.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/msvc80.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/mfplat.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/hhctrl.ocx
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/qedit.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/mscoree.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/mp3dmod.dll
./snap/notepad-plus-plus/common/.wine/drive_c/windows/syswow64/d3dx9_41.dll
```

Организуем вложенную команду.

```
lera2003@Valeriya:~/Documents$ find $(ls | grep "txt")
file_links.txt
laba2_1.txt
laba2_2.txt
laba2_3.txt
laba2.txt
script1.txt
```



6. Проанализируем содержимое заголовка файла с помощью утилиты `od`.

`Octal Dump (od)` — это команда в Linux, используемая для преобразования файловых данных в различные форматы с восьмеричным форматом по умолчанию.

```
lera2003@Valeriya:~/Documents$ echo "This is a new file" > check_od.txt
lera2003@Valeriya:~/Documents$ od -tc check_od.txt
00000000  T h i s       i s       a       n e w       f i
00000020  l e \n
00000023
lera2003@Valeriya:~/Documents$ echo "Adding new string to file!" > check_od.txt
lera2003@Valeriya:~/Documents$ od -tc check_od.txt
00000000  A d d i n g       n e w       s t r i n
00000020  g       t o       f i l e ! \n
00000033
```

Команда `od` с ключами `-tc` выводит дамп памяти, ассоциированный с указанным файлом, побайтно в восьмеричном коде, заменяя код на символы там, где это возможно.

Проанализируем содержимое файла-каталога с помощью утилиты `df`.

```
lera2003@Valeriya:~$ df ./Documents
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda1       19525500  7988004   10522612   44% /
```

Проанализируем с помощью утилиты `od` исполняемый файл.

```
lera2003@Valeriya:~$ od a.out
00000000  042577 043114 000402 000001 000000 000000 000000 000000
00000020  000002 000076 000001 000000 002060 000100 000000 000000
00000040  000100 000000 000000 000000 014740 000000 000000 000000
00000060  000000 000000 000100 000070 000011 000100 000037 000034
00000100  000006 000000 000005 000000 000100 000000 000000 000000
00000120  000100 000100 000000 000000 000100 000100 000000 000000
00000140  000770 000000 000000 000000 000770 000000 000000 000000
00000160  000010 000000 000000 000000 000003 000000 000004 000000
00000200  001070 000000 000000 000000 001070 000100 000000 000000
00000220  001070 000100 000000 000000 000034 000000 000000 000000
00000240  000034 000000 000000 000000 000001 000000 000000 000000
00000260  000001 000000 000005 000000 000000 000000 000000 000000
00000300  000000 000100 000000 000000 000000 000100 000000 000000
00000320  003374 000000 000000 000000 003374 000000 000000 000000
00000340  000000 000040 000000 000000 000001 000000 000006 000000
```

```

lera2003@Valeriya:~$ od -tc a.out
00000000 177  E  L  F 002 001 001  \0  \0  \0  \0  \0  \0  \0  \0  \0
00000020 002  \0  >  \0 001  \0  \0  \0  0 004  @  \0  \0  \0  \0  \0
00000040 @  \0  \0  \0  \0  \0  \0  \0  \0 340 031  \0  \0  \0  \0  \0
00000060 \0  \0  \0  \0  @  \0  8  \0  \t  \0  @  \0  037  \0  034  \0
00001000 006  \0  \0  \0  \0 005  \0  \0  \0  @  \0  \0  \0  \0  \0
00001200 @  \0  @  \0  \0  \0  \0  \0  \0  @  \0  @  \0  \0  \0  \0
00001400 370 001  \0  \0  \0  \0  \0  \0  \0 370 001  \0  \0  \0  \0  \0
00001600 \b  \0  \0  \0  \0  \0  \0  \0 003  \0  \0  \0 004  \0  \0  \0
00002000 8 002  \0  \0  \0  \0  \0  \0  \0 8 002  @  \0  \0  \0  \0  \0
00002200 8 002  @  \0  \0  \0  \0  \0  \0  \0 034  \0  \0  \0  \0  \0  \0
00002400 034  \0  \0  \0  \0  \0  \0  \0 001  \0  \0  \0  \0  \0  \0
00002600 001  \0  \0  \0  \0 005  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0
00003000 \0  \0  @  \0  \0  \0  \0  \0  \0  \0  \0  \0  @  \0  \0  \0  \0
00003200 374 006  \0  \0  \0  \0  \0  \0  \0 374 006  \0  \0  \0  \0  \0
00003400 \0  \0  \0  \0  \0  \0  \0 001  \0  \0  \0 006  \0  \0  \0
00003600 020 016  \0  \0  \0  \0  \0  \0  \0 020 016  \0  \0  \0  \0  \0
00004000 020 016  \0  \0  \0  \0  \0  \0  ( 002  \0  \0  \0  \0  \0  \0
00004200 0 002  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0  \0

```

7. Изменим размер каталога, варьируя количество записей. Напишем скрипт, добавляющий и удаляющий часть файлов и выводящий размер директории в течении работы программы.

```

#!/bin/bash

#создадим новую директорию
dir=newdir
mkdir $dir
#вывод изначального размера директории
echo "start size:"
du -sh $dir

#создание новых файлов внутри директории
for ((i=0;i<=15;i++)); do
    #создадим внутри папку
    mkdir $dir/$i
    #в файлы добавляем текст, чтобы они "весили"
    touch $dir/$i/file$i.txt | echo "test" >> $dir/$i/file$i.txt
done

#вывод размера директории после добавления файлов
echo "current size:"
du -sh $dir

#удаление части файлов
for ((i=0;i<=8;i++)); do
    rm $dir/$i/file$i.txt
    rmdir $dir/$i
done

#вывод размера директории после удаления части файлов
echo "final size:"
du -sh $dir

rm -r $dir

```

Результат работы скрипта.

```
lera2003@Valeriya:~$ ~/scripts/script7.sh
start size:
4,0K    newdir
current size:
132K    newdir
final size:
60K     newdir
```

Изменим значение работы цикла for, при создании каталогов до 14.

Результат работы скрипта в данном случае.

```
lera2003@Valeriya:~$ ~/scripts/script7.sh
start size:
4,0K    newdir
current size:
124K    newdir
final size:
52K     newdir
```

Таким образом, при добавлении одного каталога и одного текстового файла с его содержимым размер директории увеличивается на 8К.

## 8. Содержимое */etc/passwd*.

```
lera2003@Valeriya:~/test$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108:./home/syslog:/bin/false
_apt:x:105:65534:./nonexistent:/bin/false
messagebus:x:106:110:./var/run/dbus:/bin/false
uidd:x:107:111:./run/uidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:117:./nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127:./var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
lera2003:x:1000:1000:Lera,,,:/home/lera2003:/bin/bash
guest-254cck:x:999:999:Guest:/tmp/guest-254cck:/bin/bash
```

Файл */etc/passwd* представляет собой текстовый файл с одной записью в строке, представляющей учетную запись пользователя.

Обычно первая строка описывает пользователя *root*, за которым следуют системные и обычные учетные записи пользователей. Новые записи добавляются в конец файла.

Каждая строка файла */etc/passwd* содержит семь полей:

Username. Строка, которую вы вводите при входе в систему. Каждое имя пользователя должно быть уникальной строкой на компьютере. Максимальная длина имени пользователя ограничена 32 символами.

Password. В старых системах Linux зашифрованный пароль пользователя хранился в файле `/etc/passwd`. В большинстве современных систем это поле имеет значение `x`, и пароль пользователя сохраняется в файле `/etc/shadow`.

UID. Идентификатор пользователя — это номер, назначенный каждому пользователю. Он используется операционной системой для обращения к пользователю.

GID. Номер идентификатора группы пользователя, относящийся к основной группе пользователя. Когда пользователь создает файл, группа файла устанавливается на эту группу. Как правило, имя группы совпадает с именем пользователя. Пользователя вторичные группы перечислены в файле `/etc/groups`.

GECOS или полное имя пользователя. Это поле содержит список значений через запятую со следующей информацией:

Полное имя пользователя или название приложения.

Номер комнаты.

Рабочий номер телефона.

Домашний телефон.

Другая контактная информация.

Home directory. Абсолютный путь к домашнему каталогу пользователя. Он содержит файлы пользователя и конфигурации. По умолчанию домашние каталоги пользователей именуются по имени пользователя и создаются в каталоге `/home`.

Login shell. Абсолютный путь к оболочке входа пользователя. Это оболочка, которая запускается, когда пользователь входит в систему. В большинстве дистрибутивов Linux оболочкой входа по умолчанию является `Bash`.

Проанализируем права доступа.

```
lera2003@Valeriya:~/test$ ls -l /etc/passwd
-rw-r--r-- 1 root root 2302 мар 15 01:43 /etc/passwd
```



Файл принадлежит root, доступен для чтения всем пользователям, а для записи только root.

Содержимое */etc/shadow*.

```
lera2003@Valeriy:~/test$ sudo cat /etc/shadow
[sudo] password for lera2003:
root:$6$TxZoUAjf$mjICcYX.u6bLpk9iE8WSt9LxhqS5YICwHgVLQ.DG891FYjld08rhQa/D8kcFCNurdhczuv5GFVNVRe/NB3809.:19429:0:99999:7:::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
lp:*:18480:0:99999:7:::
mail:*:18480:0:99999:7:::
news:*:18480:0:99999:7:::
uucp:*:18480:0:99999:7:::
proxy:*:18480:0:99999:7:::
www-data:*:18480:0:99999:7:::
backup:*:18480:0:99999:7:::
list:*:18480:0:99999:7:::
irc:*:18480:0:99999:7:::
gnats:*:18480:0:99999:7:::
nobody:*:18480:0:99999:7:::
systemd-timesync:*:18480:0:99999:7:::
systemd-network:*:18480:0:99999:7:::
systemd-resolve:*:18480:0:99999:7:::
systemd-bus-proxy:*:18480:0:99999:7:::
syslog:*:18480:0:99999:7:::
_apt:*:18480:0:99999:7:::
messagebus:*:18480:0:99999:7:::
uidd:*:18480:0:99999:7:::
lightdm:*:18480:0:99999:7:::
whoopsie:*:18480:0:99999:7:::
avahi-autoipd:*:18480:0:99999:7:::
avahi:*:18480:0:99999:7:::
dnsmasq:*:18480:0:99999:7:::
colord:*:18480:0:99999:7:::
speech-dispatcher:!:18480:0:99999:7:::
hplip:*:18480:0:99999:7:::
kernoops:*:18480:0:99999:7:::
pulse:*:18480:0:99999:7:::
rtkit:*:18480:0:99999:7:::
saned:*:18480:0:99999:7:::
usbmux:*:18480:0:99999:7:::
lera2003:$6$C2pAn1p0$1saVRgT103lHxH2g79gB0dYl8tHnpEjfez3rZgGndKd99kXA6bP0Yl62c41nhd7GN4DcfoADezV/1L97d9kNj0:19429:0:99999:7:::
guest-254ckk:!:19430:0:99999:7:::
```

Так как файл доступен для чтения только root, был совершен вход в качестве супер пользователя с помощью *sudo*.

*/etc/shadow* — это текстовый файл, содержащий информацию о паролях пользователей системы.

Файл содержит следующую информацию:

Имя пользователя. Строка, которую вы вводите при входе в систему.  
Учетная запись пользователя, которая существует в системе.

Зашифрованный пароль. Пароль использует формат *\$type\$salt\$hashed*.  
*\$type* является методом криптографического алгоритма хеширования и может иметь следующие значения:

\$1\$ — MD5

\$2a\$ — Blowfish

\$2y\$ — Eksblowfish

\$5\$ — SHA-256

## \$6\$ — SHA-512

Если поле пароля содержит звездочку ( \*) или восклицательный знак ( !), пользователь не сможет войти в систему с использованием аутентификации по паролю. Другие методы входа, такие как аутентификация на основе ключей или переключение на пользователя, по-прежнему разрешены.

В старых системах Linux зашифрованный пароль пользователя хранился в файле /etc/passwd.

Последнее изменения пароля. Это дата последнего изменения пароля. Количество дней исчисляется с 1 января 1970 года (дата эпохи).

Минимальный срок действия пароля. Количество дней, которое должно пройти, прежде чем пароль пользователя может быть изменен. Как правило, он установлен на ноль, что означает отсутствие минимального срока действия пароля.

Максимальный срок действия пароля. Количество дней после смены пароля пользователя. По умолчанию этот номер установлен на 99999.

Период предупреждения. Количество дней до истечения срока действия пароля, в течение которого пользователь получает предупреждение о необходимости изменения пароля.

Период бездействия. Количество дней после истечения срока действия пароля пользователя до отключения учетной записи пользователя. Обычно это поле пустое.

Срок хранения. Дата, когда учетная запись была отключена. Это представляется как дата эпохи.

Неиспользованный. Это поле игнорируется. Оно зарезервировано для будущего использования.

Проанализируем права доступа.

```
lera2003@Valeriya:~/test$ sudo ls -l /etc/shadow
-rw-r----- 1 root shadow 1373 мар 15 01:43 /etc/shadow
```

Владельцем является root и он имеет право на чтение и запись без выполнения. Для группы право только для чтения.



Изучим утилиту `/usr/bin/passwd` – она позволяет менять пароль пользователя.

```
lera2003@Valeriya:~/test$ /usr/bin/passwd
Changing password for lera2003.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
Sorry, passwords do not match
passwd: Authentication token manipulation error
passwd: password unchanged
lera2003@Valeriya:~/test$ /usr/bin/passwd
Changing password for lera2003.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
```

Проанализируем права доступа.

```
lera2003@Valeriya:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 54256 map 26 2019 /usr/bin/passwd
```

Для того, чтобы обычный пользователь мог изменить свой пароль, существует утилита `/usr/bin/passwd`, которая доступна администратору и группе `root` на чтение, запись и исполнение, а всем остальным – на чтение и исполнение. Эта программа может выполнять действия от имени администратора, независимо от того, кто её запустил.

9. Исследуем права владения и доступа, а также их сочетаемость.

В Unix каждому файлу соответствует набор прав доступа, представленный в виде 9-ти битов режима. Он определяет, какие пользователи имеют право читать файл, записывать в него данные или выполнять его.

Для назначения прав используются три группы флагов, первая определяет права для владельца, вторая - права для основной группы пользователя, третья - для всех остальных пользователей в системе.

Для файлов: `r` - право на чтение из файла; `w` - разрешает запись в файл (в частности перезапись или изменение); `x` - позволяет исполнить файл.

Для каталогов, флаги `r w x` имеют несколько отличный смысл: `r` - позволяет читать только имена файлов в каталоге; `x` - позволяет иметь доступ к самим файлам и их атрибутам (но не именам); `w` имеет смысл только в сочетании с `x`, и позволяет (в дополнение к `x`) манипулировать с файлами в

каталоге (создавать, удалять и переименовывать). w без x - не имеет никакого эффекта.

- --- - нет прав, совсем;
- --x - разрешено только выполнение файла, как программы, но не изменение и не чтение;
- -w- - разрешена только запись и изменение файла;
- -wx - разрешено изменение и выполнение, но в случае с каталогом, вы не можете посмотреть его содержимое;
- r-- - права только на чтение;
- r-x - только чтение и выполнение, без права на запись;
- rw- - права на чтение и запись, но без выполнения;
- rwx - все права;

9.1. Создадим каталог research и применим утилиты chmod, chown.

chmod (от англ. change mode) — команда для изменения прав доступа к файлам и каталогам, используемая в Unix-подобных операционных системах.

744 - разрешить все для владельца, а остальным только чтение.

```
lera2003@Valeriya:~$ ls -l | grep 'research'
drwxrwxr-x 3 lera2003 lera2003 4096 мар 16 04:59 research
lera2003@Valeriya:~$ chmod 744 ./research
lera2003@Valeriya:~$ ls -l | grep 'research'
drwxr--r-- 3 lera2003 lera2003 4096 мар 16 04:59 research
```

Команда chown позволяет изменить владельца пользователя и / или группы для данного файла, каталога или символической ссылки.

Изменим владельца папки research на root:

```
lera2003@Valeriya:~$ sudo chown root ./research
[sudo] password for lera2003:
lera2003@Valeriya:~$ ls -l | grep 'research'
drwxr--r-- 3 root lera2003 4096 мар 16 04:59 research
```

9.2. Расширим права исполнения экспериментального файла с помощью флага SUID.

4555 (-r-sr-xr-x) SUID Каждый пользователь имеет право читать и запускать на выполнение с правами владельца файла.

```
lera2003@Valeriya:~/research$ ls -l file.txt
-rw-rw-r-- 1 lera2003 lera2003 0 map 16 04:58 file.txt
lera2003@Valeriya:~/research$ sudo chmod u+s file.txt
lera2003@Valeriya:~/research$ ls -l file.txt
-rwSrwx-r-- 1 lera2003 lera2003 0 map 16 04:58 file.txt
```

Буква S сигнализирует о флаге SUID в правах пользователя.

Другие виды расширенных прав доступа:

Working with SUID, SGID, and Sticky Bit				
Permission	Numerical Value	Relative Value	On Files	On Directories
SUID	4	u+s	User executes file with permissions of file owner.	No meaning.
SGID	2	g+s	User executes file with permissions of group owner.	File created in directory gets the same group owner.
Sticky bit	1	+t	No meaning.	Users are prevented from deleting files from other users.

9.3. Экспериментально установить, как формируются итоговые права на использование файла, если права пользователя и группы, в которую он входит, различны.

Создадим файл testing и рассмотрим пользователя и группу lera2003.

```
lera2003@Valeriya:~/research$ touch testing
lera2003@Valeriya:~/research$ echo "hi!" > testing
lera2003@Valeriya:~/research$ cat testing
hi!
lera2003@Valeriya:~/research$ ls -l testing
-rw-rw-r-- 1 lera2003 lera2003 4 map 16 05:18 testing
```

Все действия выполнены успешно. Теперь исключим все права у пользователя и дадим их все группе.

```
lera2003@Valeriya:~/research$ sudo chmod u-rwx testing
lera2003@Valeriya:~/research$ sudo chmod g+rwx testing
lera2003@Valeriya:~/research$ ls -l testing
---rwxr-- 1 lera2003 lera2003 4 map 16 05:18 testing
lera2003@Valeriya:~/research$ cat testing
cat: testing: Permission denied
lera2003@Valeriya:~/research$ echo "mew string for adding" > testing
bash: testing: Permission denied
```

После выполнения операций владелец не может что-либо сделать, не смотря на принадлежность в группе, так как ему чтение, запись и исполнение запрещены.

Подведём итог. Права доступа работают следующим образом:

- Оболочка проверяет, являетесь ли вы владельцем файла, к которому вы хотите получить доступ. Если вы являетесь этим владельцем, вы получаете разрешения и оболочка прекращает проверку.
- Если вы не являетесь владельцем файла, оболочка проверит, являетесь ли вы участником группы, у которой есть разрешения на этот файл. Если вы являетесь участником этой группы, вы получаете доступ к файлу с разрешениями, которые для группы установлены, и оболочка прекратит проверку.
- Если вы не являетесь ни пользователем, ни владельцем группы, вы получаете права других пользователей (Other)

9.4. Сопоставим возможности исполнения наиболее часто используемых операций, варьируя правами доступа к файлу и каталогу.

Если убрать права у пользователя, то базовые операции недоступны, как для файлов, так и для каталогов.

Уберем права у владельца, оставив у группы:

```
lera2003@Valeriya:~$ ls -l | grep "pravo"
drwxrwxr-x 2 lera2003 lera2003 4096 мар 23 12:59 pravo
lera2003@Valeriya:~$ chmod u-gwx pravo
lera2003@Valeriya:~$ ls -l | grep "pravo"
d--rwxr-x 2 lera2003 lera2003 4096 мар 23 12:59 pravo
lera2003@Valeriya:~$ cd pravo
bash: cd: pravo: Отказано в доступе
```

Если же у владельца права есть – операции доступны.

```
lera2003@Valeriya:~$ chmod u+gwx pravo
lera2003@Valeriya:~$ chmod g-gwx pravo
lera2003@Valeriya:~$ cd pravo
lera2003@Valeriya:~/pravo$
```

Для команды `cat` необходимы права для файла - `r`, для каталога - `x`. Для записи в файл через поток (например `cat>>`) права для файла - `w`, для каталога - `x`. Для выполнения скрипта права для файла - `rx`, для каталога - `x`. Для команды `cd` права для файла - `-`, для каталога - `x`. Для команды `ls` права для файла - `-`, для каталога - `r`.

10. Разработаем «программу-шлюз» для доступа к файлу другого пользователя при отсутствии прав на чтение информации из этого файла.

```
#include <iostream>
#include <fstream>

int main(){
    //получаем доступ к файлу
    std::ifstream file("./secret_file.txt");
    //проверяем, удалось ли открыть файл
    if (file.is_open()){
        std::string pswd;
        //считываем текст файла
        std::getline(file, pswd);
        //вывод содержимого файла
        std::cout << pswd << std::endl;
    }
    else{
        //вывод информации, что открыть не удалось
        std::cout << "access denied" << std::endl;
    }
    //закрываем фай
    file.close();
    return 0;
}
```

Командой `chmod +s ./a.out` дадим исполняемому файлу программы-шлюз разрешение SUID. Это позволит прочитать файл, если мы войдем в систему как другой пользователь.

```
lera2003@Valeriya:~/research$ g++ task10.cpp
lera2003@Valeriya:~/research$ sudo chmod +s ./a.out
lera2003@Valeriya:~/research$ ls -l ./a.out
-rwsrwsr-x 1 lera2003 lera2003 14512 мар 16 05:51 ./a.out
```

У файла `secret_file.txt` уберем права чтения у пользователей и сменим текущего пользователя.

```
lera2003@Valeriya:~/research$ ls -l secret_file.txt
-rw-rw-r-- 1 lera2003 lera2003 0 мар 16 05:52 secret_file.txt
```

```
lera2003@Valeriya:~/research$ chmod o-r secret_file.txt
lera2003@Valeriya:~/research$ ls -l secret_file.txt
-rw-rw---- 1 lera2003 lera2003 35 мар 16 05:53 secret_file.txt
```

```
lera2003@Valeriya:~$ su testinguser
Password:
testinguser@Valeriya:/home/lera2003$ ls
```

Попробуем прочитать данный файл без помощи программы и с ней.

```
testinguser@Valeriya:/home/lera2003/research$ cat secret_file.txt
cat: secret_file.txt: Permission denied
testinguser@Valeriya:/home/lera2003/research$ ./a.out
This is a very secret information!
```

Таким образом, за счёт того, что исполняемый файл имеет SUID, мы можем прочитать файл.

11. Применяя утилиту `df` и аналогичные ей по функциональности утилиты, а также информационные файлы типа `fstab`, получим информацию о файловых системах, возможных для монтирования, а также установленных на компьютере реально.

11.1. Информация об исследованных утилитах и информационных файлах с анализом их содержимого и форматов.

Утилита `df` предназначена для получения информации о свободном дисковом пространстве. Синтаксис: `df [-ghknP] [device|directory|file]`. По умолчанию дисковое пространство измеряется в количестве 512-байтных блоков. Опция `-k` используется для измерения пространства количеством 1024-байтных блоков. Опция `-h` используется для измерения пространства в единицах, удобных для чтения человеком (байты, килобайты и т.д.). Опция `-P` используется для отображения заголовков столбцов таблицы. Опция `-g` используется для отображения более подробной информации.

```
lera2003@Valeriya:~$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev            473472         0     473472   0% /dev
tmpfs           100848        6308     94540   7% /run
/dev/sda1       19525500 7989060 10521556  44% /
tmpfs           504232        352     503880   1% /dev/shm
tmpfs           5120          4       5116    1% /run/lock
tmpfs           504232         0     504232   0% /sys/fs/cgroup
/dev/loop0       128          128         0 100% /snap/bare/5
/dev/loop1       64896        64896         0 100% /snap/core20/1828
/dev/loop6       51072        51072         0 100% /snap/snapd/18357
/dev/loop2       6272         6272         0 100% /snap/notepad-plus-plus/386
/dev/loop7       354688       354688         0 100% /snap/gnome-3-38-2004/119
/dev/loop3       474624       474624         0 100% /snap/wine-platform-runtime-core20/57
/dev/loop5       93952        93952         0 100% /snap/gtk-common-themes/1535
/dev/loop4       464000       464000         0 100% /snap/wine-platform-7-devel-core20/24
none            504232       1384     502848   1% /tmp/guest-78wkxm
tmpfs           100848         72     100776   1% /run/user/998
tmpfs           100848         48     100800   1% /run/user/1000
```

Каждая строка включает следующие столбцы:

- «Файловая система» — имя файловой системы.
- «1К-блоки» — размер файловой системы в 1К-блоках.



- «Использовано» — используемое пространство в блоках по 1К.
- «Доступно» — доступное пространство в блоках по 1К.
- «Use%» — процент используемого пространства.
- «Смонтировано» — каталог, в котором смонтирована фс.

Были найдены следующие файловые системы:

- `udev` - управление устройствами для новых версий ядра Linux, являющийся преемником `devfs`, `hotplug` и `HAL`. Его основная задача — обслуживание файлов устройств (англ. `device nodes`) в каталоге `/dev` и обработка всех действий, выполняемых в пространстве пользователя при добавлении/отключении внешних устройств, включая загрузку `firmware`.

- `tmpfs` - временное файловое хранилище во многих Unix-подобных ОС. Предназначена для монтирования файловой системы, но размещается в ОЗУ вместо физического диска. Подобная конструкция является подобной RAM-диску.

- `/dev/sda` – термин `sd` означает диск SCSI, то есть диск с интерфейсом малой компьютерной системы. Итак, `sda` означает первый жесткий диск SCSI. Аналогично, `/ hda`, отдельный раздел на диске принимает имена `sda1`, `sda2` и т. Д. Активный раздел обозначается звездочкой `*` в среднем столбце.

- `/dev/loop` - это виртуальное `/ loop` устройство, которое в основном представляет собой файл в системе Linux, который действует как блочное устройство. Он используется для монтирования образов дисков, таких как `Snap`. `/dev/loop` читается только для чтения, поэтому размер фиксируется после создания и не может быть изменен. Для каждого программного обеспечения, устанавливаемого с помощью `snap`, создается устройство цикла. После удаления приложения автоматически удаляется связанный цикл `/dev/loop`.

Утилита `cfdisk` подобна `df`. Это консольная программа для разметки жесткого диска и создания на нём разделов.



```

Disk: /dev/sda
Size: 20 GiB, 21474836480 bytes, 41943040 sectors
Label: dos, identifier: 0x4958635e

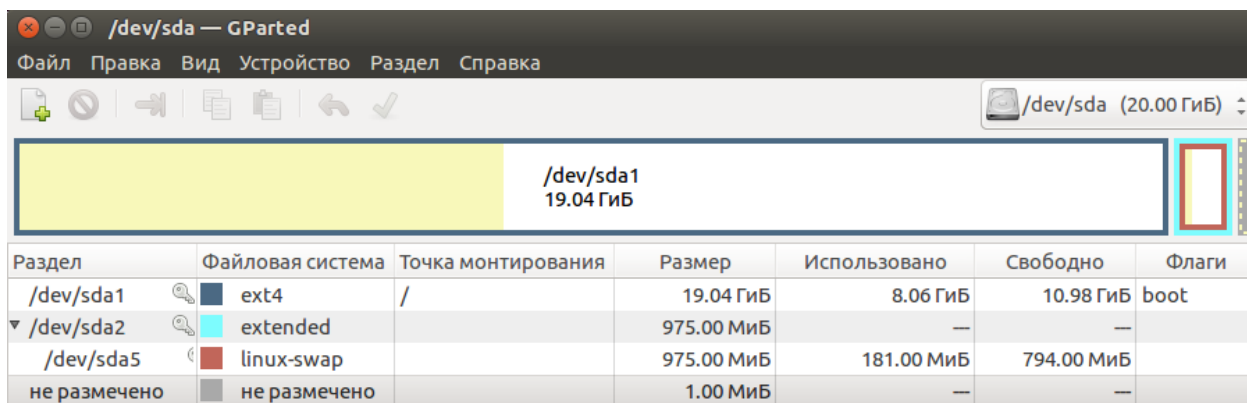
>>  Device      Boot      Start      End  Sectors  Size  Id Type
    /dev/sda1    *          2048    39942143 39940096   19G   83 Linux
    /dev/sda2          39944190 41940991 1996802   975M    5 Extended
    └─/dev/sda5          39944192 41940991 1996800   975M   82 Linux swap / Solar

Partition type: Linux (83)
Attributes: 80
Filesystem: ext4
Filesystem UUID: f926250e-7b89-4ce5-aca2-0615a25a717b
Mountpoint: / (mounted)

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

```

GParted (GNOME Partition Editor) — редактор дисковых разделов, являющийся графической оболочкой к GNU Parted.



Используя `df -a`, получаем информацию о всех ФС, известных ядру.

```

lera2003@Valeriya:~$ df -a
Filesystem      1K-blocks    Used Available Use% Mounted on
sysfs            0            0          0   -  /sys
proc            0            0          0   -  /proc
udev           473472        0    473472   0%  /dev
devpts          0            0          0   -  /dev/pts
tmpfs           100848       6316    94532   7%  /run
/dev/sda1      19525500    7996628  10513988 44%  /
securityfs      0            0          0   -  /sys/kernel/security
tmpfs           504232       352    503880   1%  /dev/shm
tmpfs           5120         4      5116   1%  /run/lock
tmpfs           504232        0    504232   0%  /sys/fs/cgroup
cgroup           0            0          0   -  /sys/fs/cgroup/systemd
pstore           0            0          0   -  /sys/fs/pstore
cgroup           0            0          0   -  /sys/fs/cgroup/hugetlb
cgroup           0            0          0   -  /sys/fs/cgroup/pids
cgroup           0            0          0   -  /sys/fs/cgroup/net_cls,net_prio
cgroup           0            0          0   -  /sys/fs/cgroup/memory
cgroup           0            0          0   -  /sys/fs/cgroup/freezer
cgroup           0            0          0   -  /sys/fs/cgroup/devices
cgroup           0            0          0   -  /sys/fs/cgroup/blkio

```

Для получения информации о реальных файловых системах на жестком диске: `df -x tmpfs`.

```
lera2003@Valeriya:~$ df -x tmpfs
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             473472         0    473472   0% /dev
/dev/sda1       19525500 7996628   10513988  44% /
/dev/loop0         128         128         0 100% /snap/bare/5
/dev/loop1        64896     64896         0 100% /snap/core20/1828
/dev/loop6        51072     51072         0 100% /snap/snapd/18357
/dev/loop2         6272      6272         0 100% /snap/notepad-plus-plus/386
/dev/loop7       354688   354688         0 100% /snap/gnome-3-38-2004/119
/dev/loop3       474624   474624         0 100% /snap/wine-platform-runtime-core20/57
/dev/loop5        93952     93952         0 100% /snap/gtk-common-themes/1535
/dev/loop4       464000   464000         0 100% /snap/wine-platform-7-devel-core20/24
```

### fstab.

Тип файла - обычный файл, разрешения для владельца - чтение и запись, разрешение для группы и для всех остальных - только чтение. fstab - file systems table - один из конфигурационных файлов в UNIX - подобных системах, содержит информацию о ФС и устройствах хранения информации компьютера.

```
lera2003@Valeriya:~$ ls -l /etc/fstab
-rw-rw-r-- 1 root root 594 map 14 01:49 /etc/fstab
```

file\_system — сообщает mount, что монтировать.

```
lera2003@Valeriya:~$ cat /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>
# / was on /dev/sda1 during installation
UUID=f926250e-7b89-4ce5-aca2-0615a25a717b /          ext4      errors=remount-ro 0          1
# swap was on /dev/sda5 during installation
UUID=dec9ad5d-cb23-498a-88b4-a2bb59dcfa7b none        swap      sw          0          0
```

11.2. Привести образ диска с точки зрения состава и размещения всех ФС на испытуемом компьютере, а также образ полного дерева ФС, включая присоединенные ФС съемных и несъемных носителей. Проанализировать и указать формат таблицы монтирования.

mount — утилита командной строки в UNIX-подобных операционных системах. Применяется для монтирования файловых систем.

```
lera2003@Valeriya:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=473472k,nr_inodes=118368,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=100848k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
```

Файл /etc/mtab — это файл, содержащий список смонтированных файловых систем. Любая смонтированная файловая система будет отображаться здесь. Если у вас есть диск или том, который не смонтирован, он не будет отображаться в этом файле.

```
lera2003@Valeriya:~$ cat /etc/mtab
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=473472k,nr_inodes=118368,mode=755 0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=100848k,mode=755 0 0
/dev/sda1 / ext4 rw,relatime,errors=remount-ro,data=ordered 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
tmpfs /sys/fs/cgroup tmpfs ro,nosuid,nodev,noexec,mode=755 0 0
cgroup /sys/fs/cgroup/systemd cgroup rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups
pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
cgroup /sys/fs/cgroup/hugetlb cgroup rw,nosuid,nodev,noexec,relatime,hugetlb 0 0
cgroup /sys/fs/cgroup/pids cgroup rw,nosuid,nodev,noexec,relatime,pids 0 0
cgroup /sys/fs/cgroup/net_cls,net_prio cgroup rw,nosuid,nodev,noexec,relatime,net_cls,net_prio 0 0
cgroup /sys/fs/cgroup/memory cgroup rw,nosuid,nodev,noexec,relatime,memory 0 0
cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,freezer 0 0
cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,devices 0 0
cgroup /sys/fs/cgroup/blkio cgroup rw,nosuid,nodev,noexec,relatime,blkio 0 0
cgroup /sys/fs/cgroup/perf_event cgroup rw,nosuid,nodev,noexec,relatime,perf_event 0 0
cgroup /sys/fs/cgroup/rdma cgroup rw,nosuid,nodev,noexec,relatime,rdma 0 0
cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,nosuid,nodev,noexec,relatime,cpu,cpuacct 0 0
cgroup /sys/fs/cgroup/cpuset cgroup rw,nosuid,nodev,noexec,relatime,cpuset 0 0
systemd-1 /proc/sys/fs/binfmt_misc autofs rw,relatime,fd=36,prgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=36
mqueue /dev/mqueue mqueue rw,relatime 0 0
hugetlbfs /dev/hugepages hugetlbfs rw,relatime,pagesize=2M 0 0
debugfs /sys/kernel/debug debugfs rw,relatime 0 0
fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
configfs /sys/kernel/config configfs rw,relatime 0 0
/dev/loop0 /snap/bare/5 squashfs ro,nodev,relatime 0 0
/dev/loop1 /snap/core20/1828 squashfs ro,nodev,relatime 0 0
```

```
lera2003@Valeriya:~$ cat /etc/mtab | grep /dev/sda
/dev/sda1 / ext4 rw,relatime,errors=remount-ro,data=ordered 0 0
```

dev / sda - первый SCSI-диск с адресным идентификатором SCSI.

SCSI (англ. Small Computer System Interface) представляет собой набор стандартов для физического подключения и передачи данных между компьютерами и периферийными устройствами.

```
lera2003@Valeriya:~$ cat /etc/mtab | grep /dev/sdb  
/dev/sdb1 /media/lera2003/Transcend vfat rw,nosuid,nodev,relatime,uid=1000,gid=1000,  
shortname=mixed,showexec=utf8,flush,errors=remount-ro 0 0
```

dev / sdb - второй SCSI-диск по адресу.

Файл fstab - это текстовый файл, который содержит информацию о различных файловых системах и устройствах хранения информации в вашем компьютере. Это всего лишь один файл, определяющий, как диск и/или раздел будут использоваться и как будут встроены в остальную систему.

```
lera2003@Valeriya:~$ cat /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>  
# / was on /dev/sda1 during installation  
UUID=f926250e-7b89-4ce5-aca2-0615a25a717b / ext4 errors=remount-ro 0 1  
# swap was on /dev/sda5 during installation  
UUID=dec9ad5d-cb23-498a-88b4-a2bb59dcfa7b none swap sw 0 0
```

Строки файла содержат следующие поля:

- Что монтируем — некоторое **блочное устройство**, которое должно быть примонтировано.
- Куда монтируем — **точка монтирования** - путь в корневой файловой системе к каталогу, в который будет смонтировано устройство
- Тип **файловой системы** монтируемого раздела
- Опции монтирования файловой системы
- Индикатор необходимости делать резервную копию (как правило не используется и равно 0)
- Порядок проверки раздела (0- не проверять, 1 - устанавливается для корня, 2 - для остальных разделов).

```
lera2003@Valeriya:~$ sudo fdisk -l
[sudo] password for lera2003:
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 63,3 MiB, 66359296 bytes, 129608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 6,1 MiB, 6352896 bytes, 12408 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 463,5 MiB, 485990400 bytes, 949200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

fdisk высвечивает информацию о разделах диска. Кроме того, fdisk создает и удаляет разделы диска, меняет активный раздел. fdisk представляет расширенный набор функций по сравнению с одноименной командой MS-DOS. fdisk обычно используется с меню в интерактивном режиме.

```
lera2003@Valeriya:~$ sudo fdisk -l | grep /dev/sda
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
/dev/sda1 *      2048 39942143 39940096   19G 83 Linux
/dev/sda2      39944190 41940991 1996802   975M  5 Extended
/dev/sda5      39944192 41940991 1996800   975M 82 Linux swap / Solaris

lera2003@Valeriya:~$ sudo fdisk -l | grep /dev/sdb
Диск /dev/sdb: 15 GiB, 16055795712 байтов, 31358976 секторов
```

Утилита fdisk позволяет посмотреть информацию о дисках, их разделах, промежутки занимаемых разделом секторов, размер блока каждого диска. Информация о флэш накопителе также отображается.

11.3. Привести «максимально возможное» дерево ФС, проанализировать, где это указывается

В файле /usr/include/linux/limits.h определена максимальная длина пути.



```

lera2003@Valeriya:~/Pictures$ cat /usr/include/linux/limits.h
#ifndef _LINUX_LIMITS_H
#define _LINUX_LIMITS_H

#define NR_OPEN          1024

#define NGROUPS_MAX      65536 /* supplemental group IDs are available */
#define ARG_MAX          131072 /* # bytes of args + environ for exec() */
#define LINK_MAX         127 /* # links a file may have */
#define MAX_CANON        255 /* size of the canonical input queue */
#define MAX_INPUT        255 /* size of the type-ahead buffer */
#define NAME_MAX         255 /* # chars in a file name */
#define PATH_MAX         4096 /* # chars in a path name including nul */
#define PIPE_BUF         4096 /* # bytes in atomic write to a pipe */
#define XATTR_NAME_MAX   255 /* # chars in an extended attribute name */
#define XATTR_SIZE_MAX   65536 /* size of an extended attribute value (64k) */
#define XATTR_LIST_MAX   65536 /* size of extended attribute namelist (64k) */

#define RTSIG_MAX        32

#endif

```

Максимально допустимая длина имени файла – 255 байт.

4096 байт – это длина полного пути до файла, включая имя.

```

lera2003@Valeriya:~/Documents$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1       7:1      0   6,1M  1 loop /snap/notepad-plus-plus/386
loop6       7:6      0 463,5M  1 loop /snap/wine-platform-runtime-core20/57
loop4       7:4      0 346,3M  1 loop /snap/gnome-3-38-2004/119
sr0         11:0     1 1024M  0 rom
loop2       7:2      0  63,3M  1 loop /snap/core20/1828
loop0       7:0      0    4K   1 loop /snap/bare/5
loop7       7:7      0  49,9M  1 loop /snap/snapd/18357
sda         8:0      0   20G   0 disk
├─sda2      8:2      0    1K   0 part
├─sda5      8:5      0  975M   0 part [SWAP]
└─sda1      8:1      0   19G   0 part /
loop5       7:5      0  91,7M  1 loop /snap/gtk-common-themes/1535
loop3       7:3      0 453,1M  1 loop /snap/wine-platform-7-devel-core20/24

```

```
Disk /dev/loop0: 4 KiB, 4096 bytes, 8 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop1: 6,1 MiB, 6352896 bytes, 12408 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop2: 63,3 MiB, 66359296 bytes, 129608 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop3: 453,1 MiB, 475066368 bytes, 927864 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop4: 346,3 MiB, 363151360 bytes, 709280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop5: 91,7 MiB, 96141312 bytes, 187776 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop6: 463,5 MiB, 485990400 bytes, 949200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/loop7: 49,9 MiB, 52260864 bytes, 102072 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```



```

Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
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: dos
Disk identifier: 0x4958635e

Device      Boot      Start      End  Sectors  Size Id Type
/dev/sda1   *          2048 39942143 39940096   19G 83 Linux
/dev/sda2                39944190 41940991 1996802   975M  5 Extended
/dev/sda5                39944192 41940991 1996800   975M 82 Linux swap / Solaris

```

### Вывод.

Отображены название диска, его размер и количество секторов.

На моем устройстве присутствует один жесткий диск – dev/sda, который содержит 3 раздела: dev/sda1, dev/sda2, dev/sda5.

Файловые системы частей следующие: sda1 – ext4, sda2 - отсутствует, sda5 – swap. Типы разделов (под что они отведены): sda1 - Linux, sda2 - Extended, sda5 – Linux swap / Solaris.

Присутствует 8 устройств типа loop – виртуальные устройства, позволяющие делать компьютер доступным в качестве блочного устройства. Они не имеют никакого отношения к занятию оперативной памяти. Обычно они используются для монтирования образов дисков.

### 12. Проанализировать и пояснить принцип работы утилиты file.

Утилита определяет тип файла. Для этого она выполняет разные тесты, которые можно разделить на 3 группы:

Filesystem tests – основаны на анализе кода возврата системного вызова stat(). Программа проверяет не пустой ли файл, и не принадлежит ли он к одному из специальных типов файлов. Все известные типы файлов распознаются, если они определены в системном файле /usr/include/sys/stat.h.

Magic number tests – используются для проверки файлов, данные в которых записаны в определённом формате. В определённом месте в начале таких файлов записано магическое число, которое позволяет ОС определить тип файла. Все известные ОС магические числа по умолчанию хранятся в файле /usr/share/misc/magic.

Language tests – используются для анализа языка, на котором написан файл, если это файл в формате ASCII. Выполняется поиск стандартных строк, которые могут соответствовать определённому языку.

Первый тест, который завершится успешно, выводит тип файла. Типы файлов можно разделить на 3 основные группы:

Текстовые – файл содержит только ASCII символы и может быть безопасно прочитан на терминале.

Исполняемые – файл содержит результаты компилирования программы в форме понятной ядру ОС.

Данные – всё, что не подходит в первые 2 группы (обычно это бинарные или непечатаемые файлы). Исключение составляют well-known форматы, используемые для хранения бинарных данных.

12.1. Привести алгоритм её функционирования на основе информационной базы, размещение и полное имя которой указывается в описании утилиты в технической документации ОС (как правило, /usr/share/file/magic.\*), а также содержимого заголовка файла, к которому применяется утилита. Определить, где находятся магические числа и иные характеристики, идентифицирующие тип файла, применительно к исполняемым файлам, а также файлам других типов.

Информация о тестах на магические числа содержится в файле /usr/share/mime/magic.

```

lera2003@Valeriya:~$ cat /usr/share/mime/magic
MIME-Magic
[90:application/vnd.stardivision.writer]
>2089=
StarWriter
[90:application/x-docbook+xml]
>0=<?xml
1>0=0017//OASIS//DTD DocBook XML+101
1>0=0017//KDE//DTD DocBook XML+101
[90:image/x-eps]
>0=0024!
1>15=0024PS
>0=0024!
1>16=0024PS
>0=0024
[80:application/prs.plucker]
>60DataPlkr
[80:application/vnd.corel-draw]
>8CDRXvrsn&00000000
[80:application/x-fictionbook+xml]
>0=
<FictionBook+257
[80:application/x-mobipocket-ebook]
>60BOOKMOBI

```

12.2. Утилиту file выполнить с разными ключами.

```

lera2003@Valeriya:~/Pictures$ file forest.jpg
forest.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=6], baseline, precision 8, 482x361, frames 3

```

-b, --brief — запрет на демонстрацию имен и адресов файлов в выводе команды;

```

lera2003@Valeriya:~/Pictures$ file -b forest.jpg
JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=6], baseline, precision 8, 482x361, frames 3

```

-i, --mime — определение MIME-типа документа по его заголовку;

```

lera2003@Valeriya:~/Pictures$ file -i forest.jpg
forest.jpg: image/jpeg; charset=binary

```

--mime-type, --mime-encoding — определение конкретного элемента MIME;

```

lera2003@Valeriya:~/Pictures$ file --mime-type forest.jpg
forest.jpg: image/jpeg

```

-l, --list — список паттернов и их длина;

```
lera2003@Valeriya:~/Pictures$ file -l forest.jpg
Set 0:
Binary patterns:
Text patterns:
Set 1:
Binary patterns:
Text patterns:
Set 0:
Binary patterns:
Strength = 380@6: OpenSSH private key []
Strength = 361@66: EICAR virus test files []
Strength = 340@585: sc68 Atari ST music []
Strength = 340@35: T64 tape Image []
Strength = 340@40: T64 tape Image []
Strength = 340@19: Erlang JAM file - version 4.3 []
Strength = 340@51: Mathematica binary file []
Strength = 340@524: %s [application/vnd.ms-excel]
Strength = 340@52: Bazaar merge directive []
Strength = 340@50: SQLite 2.x database []
Strength = 340@136: Paged COBALT boot rom []
Strength = 331@244: NetImmerse game engine file []
```

12.3. Привести экспериментальную попытку с добавлением в базу собственного типа файла и его дальнейшей идентификацией.

С помощью утилиты vim откроем файл /etc/magic и добавим новый тип файла os, который будет определено магическим число МА в своем содержании.

```
lera2003@Valeriya:~/Pictures$ sudo vim /etc/magic
[sudo] password for lera2003:
lera2003@Valeriya:~/Pictures$ cat /etc/magic
# Magic local data for file(1) command.
# Insert here your local magic data. Format is described in magic(5).
0 string MN os
```

Создадим новый файл с магическим числом и определим его тип.

```
lera2003@Valeriya:~/Pictures$ touch newtype_file
lera2003@Valeriya:~/Pictures$ echo "MN" >> newtype_file
lera2003@Valeriya:~/Pictures$ cat newtype_file
MN
lera2003@Valeriya:~/Pictures$ file newtype_file
newtype_file: os
```

Файл определяется с типом os, следовательно, новый тип был успешно создан.

## **Вывод.**

При выполнении данной лабораторной работы были изучены принципы организации файловой системы ОС Linux и получены навыки работы с ней.

Существует 6 типов файлов: обычные файлы, каталоги, символичные ссылки, блочные устройства, символичные устройства, символичные устройства, каналы и сокеты.

Были написаны следующие скрипты: выводящий все типы файлов в директории, нахождение всех жестких ссылок на заданный файл, поиск всех символических ссылок на файл, создание и удаление директорий для отслеживания выделяемой памяти.

Отработано два способа создания символических ссылок: `ln -s`, `cp -s`.

Изучена утилита `find`, которая предназначена для поиска файлов и каталогов.

Утилита `od` позволяет просмотреть дампы памяти файла в разных форматах. С помощью нее при просмотре сигнатуры объектного файла можно узнать его магическое число.

Файл `/etc/passwd` – содержит информацию об учетных записях пользователей устройства. `Etc/shadow` – информация о паролях пользовательской системы. Утилита `/usr/bin/passwd` позволяет изменить пароль.

Исследованы права владения и доступа, их сочетаемость. Используется три группы флагов: первая – права для владельца, вторая – для основной группы, третья – для остальных. Атрибут SUID, установленный на исполняемом файле, позволяет любому пользователю, который запустил этот файл, совершать действия от имени владельца файла.

Разработана программа-шлюз, для того чтобы пользователей, у которого нет прав на открытие какого-либо файла, в итоге открыл его.

Утилита `df` позволяет получить информацию о файловой системе. С помощью `fdisk` изучено пространство устройства. Выводы о том, как устроено мое устройство представлено в выводе в 11 пункте.

Утилита `file` позволяет определить тип файла. Также был создан собственный тип файла и исследована возможность его определения под данным типом.

#### **Список источников.**

1. <https://selectel.ru/blog/directory-structure-linux/>
2. [Losst - Linux Open Source Software Technologies](#)
3. [Linux - начинающим. Часть 4. Работаем с файловой системой. Теория - Записки IT специалиста \(interface31.ru\)](#)