Санкт-Петербургский Государственный Политехнический
Университет Петра Великого
Институт Компьютерных Наук и Технологий

Кафедра Компьютерных Систем и Программных Технологий

Отчет по лабораторной работе №2

по дисциплине «Операционные системы»
на тему «Файловые системы»

Выполнил студент группы 43501/3 _____ Круминьш Д.В.
(подпись)

Преподаватель _____ Душутина Е.В.
(подпись)

Санкт-Петербург
2016

1 **Ознакомиться с типами файлов исследуемой ФС. Применяя утилиту ls, отфильтровать по одному примеру каждого типа файла используемой вами ФС. Комбинируя различные ключи утилиты рекурсивно просканировать все дерево, анализируя крайнюю левую позицию выходной информации полученной посредством ls –l. Результат записать в выходной файл с указанием полного пути каждого примера. Выполнить задание сначала в консоли построчно, выбирая необходимые сочетания ключей (в командной строке), а затем оформить как скрипт с задаваемым в командной строке именем файла как параметр.**

1.1 Решение в командной строке

Исследуем структуру файловой системы на своем компьютере и назначение содержимого корневого каталога.

```
1 psaer@psaer-pc:~$ sudo ls / -lR 2>/dev/null|awk '{if($0~/^\//) path=
    ↪ substr($0,0,length($0)-1); else {  if ($0~/^l/) $(NF-2)=path"/"$(
    ↪ NF-2); else {$NF=path"/"$NF} print $0} }' | grep -v ^и|grep -v
    ↪ ^/|sort -k1.1,1.1|uniq -w1
2 ---------- 1 root root 0 сен 27 19:30 /run/crond.reboot
3 brw-rw---- 1 root disk 8, 0 сен 27 19:30 /dev/sda
4 c--------- 1 root root 5, 2 сен 27 19:29 /dev/pts/ptmx
5 d--------- 2 root root 40 сен 27 19:29 /run/systemd/inaccessible
6 lr-------- 1 psaer psaer 64 окт 1 13:07 /proc/1002/map_files/5614
    ↪ be8ff000-5614be934000 -> /usr/bin/dbus-daemon
7 prw------- 1 root root 0 окт 1 16:50 /run/systemd/inhibit/37.ref
8 srw------- 1 psaer psaer 0 сен 27 19:30 /tmp/ssh-Z2v0kmbZUTUu/agent
    ↪ .879
```

Листинг 1: Типы файлов

Разбор команды:

- sudo - необходимо для получение доступа
- ls / - указываем корневой каталог
- -lR - устанавливаем вывод полной информации и рекурсивное прохождение
- 2>/dev/null - перенаправление потока ошибок
- awk - скрипт в котором обрабатывается и переписывается название файла на полный путь с названием файла
- grep -v ^и - избавление от лишней информации
- grep -v ^/ - избавление от лишней информации
- sort -k1.1,1.1 - сортировка по первому символу
- uniq -w1 - уникальность по первому символу

Результаты:

Флаг	Описание	Тип
-	Отсутствие флага	Регулярные файлы
b	Блочное устройство	Специальные файлы
c	Символьное устройство	Специальные файлы

d	Директория	Каталог
l	Символьная ссылка	Специальные файлы
p	Канал, устройство fifo	Специальные файлы
s	Unix сокет	Специальные файлы

1.2 Script

Скрипт написан на основе awk из консольной версии.

```
#!/bin/bash
output=$1
if [ -z $output ]; then
        output="output.out"
fi
ls / -lR 2>/dev/null|awk '{if($0~/^\//) path=substr($0,0,length($0)-1)
   ↪ ; else {  if ($0~/^l/) $(NF-2)=path"/"$(NF-2); else {$NF=path"/"
   ↪ $NF} print $0} }' | grep -v ^и|grep -v ^/|sort -k1.1,1.1|uniq -w1
   ↪ >$output
```

Листинг 2: prog_2_1_2

Рассмотрим алгоритм работы awk:

1. Проверка на наличие /(слеша) в начале строки, необходимо для того чтобы запомнить путь к файлу
2. Если первый символ l(символьная ссылка), то необходимо сдвинуть место куда вставлять путь к файлу
3. Иначе заменяем имя файла на путь к файлу и имя файла

Результаты работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/work$ ls
prog_2_1_2
psaer@psaer-pc:~/Study/s7/OS/lab2/work$ ./prog_2_1_2
psaer@psaer-pc:~/Study/s7/OS/lab2/work$ cat output.out
---------- 1 root root 0 сен 27 19:30 /run/crond.rebooты
brw-rw---- 1 root disk 8, 0 сен 27 19:30 /dev/sda
c--------- 1 root root 5, 2 сен 27 19:29 /dev/pts/ptmx
d--------- 2 root root 40 сен 27 19:29 /run/systemd/inaccessible
lr-------- 1 psaer psaer 64 окт 1 13:07 /proc/1002/map_files/5614
   ↪ be8ff000-5614be934000 -> /usr/bin/dbus-daemon
prw------- 1 root root 0 окт 1 16:50 /run/systemd/inhibit/37.ref
srw------- 1 psaer psaer 0 сен 27 19:30 /tmp/ssh-Z2vOkmbZUTUu/agent
   ↪ .879
psaer@psaer-pc:~/Study/s7/OS/lab2/work$ ./prog_2_1_2 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/work$ cat temp
---------- 1 root root 0 сен 27 19:30 /run/crond.reboot
brw-rw---- 1 root disk 8, 0 сен 27 19:30 /dev/sda
c--------- 1 root root 5, 2 сен 27 19:29 /dev/pts/ptmx
d--------- 2 root root 40 сен 27 19:29 /run/systemd/inaccessible
lr-------- 1 psaer psaer 64 окт 1 13:07 /proc/1002/map_files/5614
   ↪ be8ff000-5614be934000 -> /usr/bin/dbus-daemon
prw------- 1 root root 0 окт 1 16:50 /run/systemd/inhibit/37.ref
```

```
srw------- 1 psaer psaer 0 сен 27 19:30 /tmp/ssh-Z2vOkmbZUTUu/agent
   ↪ .879
```

Листинг 3: Вывод типов файлов используя скрипт

В первом случае, запись происхдила в файл по умолчанию прописанный в скрипте, во втором случае выходной файл был указан в консоли.

2 Получить все жесткие ссылки на заданный файл, находящиеся в разных каталогах пользовательского пространства (разными способами, не применяя утилиты file и find). Использовать конвейеризацию и фильтрацию. Оформить в виде скрипта.

При написании скрипта был использован индексный дескриптор(inode) для определения жестких ссылок на указанный файл. Также имеется обработка ситуаций когда файл не введен или не найден.

```
#!/bin/bash
file=$1;
if [ -z $file ]; then
        echo "ERROR. No input file."
        exit 1
fi
inode="$(ls -i $file 2>/dev/null | awk '{print $1}')"
if [ -z $inode ]; then
        echo "ERROR. Such file does not exist."
        exit 1
fi
echo "Input file inode="$inode
ls $HOME -lRi 2>/dev/null|awk '{if($0~/^\//) path=substr($0,0,length(
   ↪ $0)-1); else {  if ($0~/^l/) $(NF-2)=path"/"$(NF-2);}'|grep ^
   ↪ $inode
exit 0
```

Листинг 4: prog_2_2

Результаты работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls
2_1_1  2_1_2  2_2  prog_2_1_2  prog_2_2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ mkdir folder1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ mkdir folder2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ln 2_1_1 2
   ↪ _1_1_copy1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ mv 2_1_1_copy1
   ↪ folder1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ln 2_1_1 2
   ↪ _1_1_copy2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ mv 2_1_1_copy2
   ↪ folder2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_2 2_1_1
7741699
Input file inode=7741699
7741699 -rw-r--r-- 5 psaer psaer 718 окт 1 19:18 /home/psaer/Study/s7/
   ↪ OS/lab2/otchet/listings/2_1_1
7741699 -rw-r--r-- 5 psaer psaer 718 окт 1 19:18 /home/psaer/Study/s7/
   ↪ OS/lab2/otchet/listings/folder1/2_1_1_copy1
```

```
7741699 -rw-r--r-- 5 psaer psaer 718 окт 1 19:18 /home/psaer/Study/s7/
   ↪ OS/lab2/otchet/listings/folder2/2_1_1_copy2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_2
ERROR. No input file.
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_2 qwerty
ERROR. Such file does not exist.
```

Листинг 5: Поиск жестких ссылок

Все созданные ссылки были успешно определены и представлены в результате.

3 Проанализировать все возможные способы формирования символьных ссылок (ln, link,cp и т.д.), продемонстрировать их экспериментально. Предложить скрипт, подсчитывающий и перечисляющий все полноименные символьные ссылки на файл, размещаемые в разных местах файлового дерева.

Утилита link позволяет создавать лишь жесткие ссылки, поэтому рассмотри работу утилит ln и cp. Утилита ln по умолчанию создает жесткие ссылки, поэтому необходимо добавить флаг -s для создания символьной ссылки. Утилита cp хоть и создана для копирования файлов, но также позволяет создавать символьную ссылку добавив флаг -s.

Пример работы ln:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls
2_1_1  2_1_2  2_2  prog_2_1_2  prog_2_2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ln -s 2_1_2 2
   ↪ _1_2_symbolic1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ln -s 2_1_2 2
   ↪ _1_2_symbolic2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l
итого 28
-rw-r--r-- 5 psaer psaer  718 окт  1 19:18 2_1_1
-rw-r--r-- 1 psaer psaer 1260 окт  1 19:19 2_1_2
lrwxrwxrwx 1 psaer psaer    5 окт  1 21:41 2_1_2_symbolic1 -> 2_1_2
lrwxrwxrwx 1 psaer psaer    5 окт  1 21:42 2_1_2_symbolic2 -> 2_1_2
-rw-r--r-- 1 psaer psaer 1146 окт  1 21:03 2_2
-rwxr-xr-x 1 psaer psaer  279 окт  1 19:01 prog_2_1_2
-rwxr-xr-x 3 psaer psaer  420 окт  1 21:06 prog_2_2
```

Листинг 6: Создание символьных ссылок используя ln

Пример работы cp:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls
2_1_1  2_1_2  2_2  2_3_ln  prog_2_1_2  prog_2_2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp -s 2_1_1 2
   ↪ _1_1_symbolic1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp -s 2_1_1 2
   ↪ _1_1_symbolic2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l
итого 24
-rw-r--r-- 5 psaer psaer  718 окт  1 19:18 2_1_1
lrwxrwxrwx 1 psaer psaer    5 окт  1 21:48 2_1_1_symbolic1 -> 2_1_1
lrwxrwxrwx 1 psaer psaer    5 окт  1 21:48 2_1_1_symbolic2 -> 2_1_1
-rw-r--r-- 1 psaer psaer 1260 окт  1 19:19 2_1_2
-rw-r--r-- 1 psaer psaer 1146 окт  1 21:03 2_2
```

```
-rw-r--r-- 1 psaer psaer  857 окт  1 21:44 2_3_ln
-rwxr-xr-x 1 psaer psaer  279 окт  1 19:01 prog_2_1_2
-rwxr-xr-x 3 psaer psaer  420 окт  1 21:06 prog_2_2
```

Листинг 7: Создание символьных ссылок используя ср

Был написан скрипт который подсчитывает количество символьных ссылок, а также вывод их местоположения относительно текущего каталога.

```
#!/bin/bash
echo "Symbolic links count:"
ls -lR | grep '\-> '$1 | wc -l
echo "Links:"
ls -lR | awk '{if($0~/^.\//) path=substr($0,0,length($0)-1); else {
   ↪ if ($0~/^l/) $(NF-2)=path"/"$(NF-2); else {$NF=path"/"$NF} print
   ↪ $0} }' |grep '\-> '$1
```

Листинг 8: prog_2_3

Результаты работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l
итого 40
-rw-r--r-- 5 psaer psaer  718 окт  1 19:18 2_1_1
-rw-r--r-- 1 psaer psaer 1260 окт  1 19:19 2_1_2
-rw-r--r-- 1 psaer psaer 1146 окт  1 21:03 2_2
-rw-r--r-- 1 psaer psaer  791 окт  1 21:48 2_3_cp
-rw-r--r-- 1 psaer psaer  731 окт  1 21:49 2_3_ln
drwxr-xr-x 3 psaer psaer 4096 окт  1 22:06 folder1
drwxr-xr-x 2 psaer psaer 4096 окт  1 21:51 folder2
-rwxr-xr-x 1 psaer psaer  279 окт  1 19:01 prog_2_1_2
-rwxr-xr-x 3 psaer psaer  420 окт  1 21:06 prog_2_2
-rwxr-xr-x 1 psaer psaer  242 окт  1 22:12 prog_2_3
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_3 2_1_1
Symbolic links count:
2
Links:
lrwxrwxrwx 1 psaer psaer 5 окт 1 21:48 ./folder1/subfolder/2
   ↪ _1_1_symbolic1 -> 2_1_1
lrwxrwxrwx 1 psaer psaer 5 окт 1 21:48 ./folder2/2_1_1_symbolic2 -> 2
   ↪ _1_1
```

Листинг 9: Поиск символьных ссылок используя скрипт

Результаты соотвествуют ожиданиям.

4 Получить все символьные ссылки на заданный в качестве входного параметра файл, не используя file (разными способами, не применяя утилиту file).

Немного изменим программу 2_3_prog, был реализован глобальный поиск, вместо поиска в текущей директории.

```
#!/bin/bash
echo "Symbolic links count:"
ls / -lR 2>/dev/null| grep '\-> '$1 | wc -l
echo "Links:"
```

```
5 ls / -lR 2>/dev/null| awk '{if($0~/^\//) path=substr($0,0,length($0)
    ↪ -1); else {  if ($0~/^l/) $(NF-2)=path"/"$(NF-2); else {$NF=path
    ↪ "/"$NF} print $0} }' |grep '\-> '$1
```

Листинг 10: prog_2_4

Результаты работы:

```
1 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo ./prog_2_4 2
    ↪ _1_1
2 Symbolic links count:
3 2
4 Links:
5 lrwxrwxrwx 1 psaer psaer 5 окт 1 21:48 /home/psaer/Study/s7/OS/lab2/
    ↪ otchet/listings/folder1/subfolder/2_1_1_symbolic1 -> 2_1_1
6 lrwxrwxrwx 1 psaer psaer 5 окт 1 21:48 /home/psaer/Study/s7/OS/lab2/
    ↪ otchet/listings/folder2/2_1_1_symbolic2 -> 2_1_1
```

Листинг 11: Глобальный поиск символьных ссылок используя скрипт

Результаты соотвествуют ожиданиям, был выведен полнуй путь к файлам.

5 Изучить утилиту find, используя ее ключи получить расширенную информацию о всех типах файлов. Создать примеры вложенных команд.

find — утилита поиска файлов по имени и другим свойствам, используемая в UNIX-подобных операционных системах. Может производить поиск в одном или нескольких каталогах с использованием критериев, заданных пользователем. По умолчанию, find возвращает все файлы в рабочей директории. Более того, find позволяет применять пользователю определённые действия ко всем найденным файлам. Также поддерживаются регулярные выражения.

Примеры возможных ключей:

```
1 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -type d
2 .
3 ./folder1
4 ./folder1/subfolder
5 ./folder2
```

Листинг 12: Поиск папок

Были выведены все файлы(каталоги) с типом d относительно текущей директории.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls
2  2_1_1  2_1_2  2_2  2_3_cp  2_3_ln  2_3_prog  2_4_prog  2_5_1  folder1
     ↪  folder2  prog_2_1_2  prog_2_2  prog_2_3  prog_2_4
3  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "2*"
4  ./2_2
5  ./2_1_2
6  ./2_3_prog
7  ./folder1/subfolder/2_1_1_symbolic1
8  ./2_3_ln
9  ./2_1_1
10 ./2_3_cp
11 ./2_4_prog
12 ./2_5_1
13 ./folder2/2_1_1_symbolic2
```

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find ~/Study/s7/OS
    ↪ -name "*.tex"
/home/psaer/Study/s7/OS/lab2/otchet/lab2.tex
/home/psaer/Study/s7/OS/lab1/extra/otchet/extra.tex
/home/psaer/Study/s7/OS/lab1/tex/os_lab1.tex
```

Листинг 13: Поиск файлов

Были выведены все файлы, которые начинаются с символа 2, затем файлы с расширением .tex в указанной папке.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l
итого 60
-rw-r--r-- 5 psaer psaer  718 окт  1 19:18 2_1_1
-rw-r--r-- 1 psaer psaer 1260 окт  1 19:19 2_1_2
-rw-r--r-- 1 psaer psaer 1146 окт  1 21:03 2_2
-rw-r--r-- 1 psaer psaer  791 окт  1 21:48 2_3_cp
-rw-r--r-- 1 psaer psaer  731 окт  1 21:49 2_3_ln
-rw-r--r-- 1 psaer psaer  868 окт  1 22:13 2_3_prog
-rw-r--r-- 1 psaer psaer  353 окт  1 22:29 2_4_prog
-rw-r--r-- 1 psaer psaer  105 окт  2 12:19 2_5_1
-rw-r--r-- 1 psaer psaer  599 окт  2 12:33 2_5_2
drwxr-xr-x 3 psaer psaer 4096 окт  1 22:06 folder1
drwxr-xr-x 2 psaer psaer 4096 окт  1 21:51 folder2
-rwxr-xr-x 1 psaer psaer  279 окт  1 19:01 prog_2_1_2
-rwxr-xr-x 3 psaer psaer  387 окт  1 22:14 prog_2_2
-rwxr-xr-x 1 psaer psaer  242 окт  1 22:12 prog_2_3
-rwxr-xr-x 1 psaer psaer  267 окт  1 22:29 prog_2_4
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -size +700c -
    ↪ size -1000c
./2_3_prog
./2_3_ln
./2_1_1
./2_3_cp
```

Листинг 14: Поиск файлов по размеру

Были выведены файлы которые больше 700, но меньше 1000 байт.

Использование утилиты find с опцией exec позволяет создавать вложенные команды. Вызов exec будет происходить из текущей директории. Все символы, следующие за командой exec, считаются ее аргументами до символа ";". Аргумент "{}"заменяется на имя рассматриваемого файла каждый раз, когда он встречается среди аргументов команды.

Примеры работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -size +700c -
    ↪ size -1000c
./2_5_3
./2_3_prog
./2_3_ln
./2_1_1
./2_3_cp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -size +700c -
    ↪ size -1000c -exec ls -l {} \;
-rw-r--r-- 1 psaer psaer 992 окт  2 12:39 ./2_5_3
-rw-r--r-- 1 psaer psaer 868 окт  1 22:13 ./2_3_prog
-rw-r--r-- 1 psaer psaer 731 окт  1 21:49 ./2_3_ln
```

```
-rw-r--r-- 5 psaer psaer 718 окт  1 19:18 ./2_1_1
-rw-r--r-- 1 psaer psaer 791 окт  1 21:48 ./2_3_cp
```

Листинг 15: Поиск файлов по размеру с подробной информацией

С помощью find были найдены файлы которые больше 700, но меньше 1000 байт, затем используя ls была выведена их более подробная информация.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "2_3*"
./2_3_prog
./2_3_ln
./2_3_cp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "2_3*" -
   ↪ exec ln -s {} {}_symb \;
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "2_3*"
./2_3_prog
./2_3_ln
./2_3_prog_symb
./2_3_cp
./2_3_cp_symb
./2_3_ln_symb
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "*_symb"
./2_3_prog_symb
./2_3_cp_symb
./2_3_ln_symb
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "*_symb"
   ↪  -exec rm {} \;
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ find -name "2_3*"
./2_3_prog
./2_3_ln
./2_3_cp
```

Листинг 16: Создание символьных ссылок и их удаление

С помощью find были найдены файлы начинающиеся на 2_3, были созданы символьные ссылки на эти файлы, затем удалены ранее созданные символьные ссылки.

6 Проанализировать содержимое заголовка файла, а также файла-каталога с помощью утилит od и *dump. Если доступ к файлу-каталогу возможен (для отдельных модификаций POSIX - совместимых ОС), проанализировать изменение его содержимого при различных операциях над элементами, входящими в его состав (файлами и подкаталогами).

Утилита od позволяет произвести вывод содержимого файла в определенном формате. В выводе утилиты od каждая строка содержит выходное смещение и следующий за ним блок данных. По умолчанию смещение выводится в восьмеричном формате.

Пример работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ touch temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ echo "123">temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rw-r--r-- 1 psaer psaer 4 окт  2 14:12 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ od temp
0000000 031061 005063
0000004
```

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ od -c temp
0000000   1   2   3  \n
0000004
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ od -bc temp
0000000 061 062 063 012
          1   2   3  \n
0000004
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ od folder1
od: folder1: ошибка чтения: Это каталог
0000000
```

Листинг 17: Утилита od

Утилита hexdump схожа по работе с утилитой od, но имеет больше возможностей при выводе.

Пример работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ hexdump -bc temp
0000000 061 062 063 064 065 066 067 070 071 060 161 167 145 162 164
   ↪ 171
0000000   1   2   3   4   5   6   7   8   9   0   q   w   e   r   t
   ↪ y
0000010 165 151 157 160 141 163 144 146 147 150 152 153 154 172 170
   ↪ 143
0000010   u   i   o   p   a   s   d   f   g   h   j   k   l   z   x
   ↪ c
0000020 166 142 156 155 012
0000020   v   b   n   m  \n
0000025
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ hexdump -v -e '"%2
   ↪ _ad | " 8 "%_c" "|\n"' temp
 0 | 12345678|
 8 | 90qwerty|
16 | uiopasdf|
24 | ghjklzxc|
32 | vbnm\n|
```

Листинг 18: Утилита hexdump

Вначале аналогично утилите od было выведено содержимое файла с указанием символов и их восьмиричных кодов. Далее этот же файл был выведен с использованием флага -e для установления формата вывода. Сперва было выведено смещение относительно начала файла, а затем установленное количество символов для вывода.

7 Определить максимальное количество записей в каталоге. Изменить размер каталога, варьируя количество записей (для этого создать программу, порождающую новые файлы и каталоги, а затем удаляющую их, предусмотрев промежуточный и конечный вывод информации о размере подопытного каталога).

По умолчанию стандартный размер каталога равен 4096 байт, имеено 4096 байт, так как это стандартный блок для моей файловый системы(ext4). Однако этот размер увеличивается по мере наполнения каталога файлами или другими каталогами. Узнаем на каком количестве вложенных файлов размер измениться, для этого был написан скрипт который наполняет каталог другими каталогами, до тех пор пока размер родительского каталога не изменится.

```
#!/bin/bash
```

```
mkdir testfolder
size1=$(ls -ld testfolder | cut -d ' ' -f5)
size2=$size1
i=0
while [ "$size1" -eq "$size2" ]
do
        mkdir ./testfolder/$i
        let "i=$i+1"
        size2=$(ls -ld testfolder | cut -d ' ' -f5)
done
rm -rf testfolder
echo "Directory size changed from $size1 to $size2 at $i dir inside"
```

Листинг 19: Скрипт по наполнению каталога

Результат работы:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_7
Directory size changed from 4096 to 12288 at 339 dir inside
```

Листинг 20: Результат наполнения каталога

По результатам работы, изменения размера каталога произошло на 339 каталогах внутри. Теперь попробуем определить максимальное количество каталогов внутри, для этого немного изменим скрипт по наполнению.

```
#!/bin/bash
mkdir testfolder
size1=$(ls -ld testfolder | cut -d ' ' -f5)
size2=$size1
i=0
while [ 1 ]
do
        mkdir ./testfolder/$i
        let "i=$i+1"
        size2=$(ls -ld testfolder | cut -d ' ' -f5)
        if [ "$size1" -ne "$size2" ]; then
                size1=$(ls -ld testfolder | cut -d ' ' -f5)
                echo "New directory size is $size1, files inside: $i"
        fi

done
```

Листинг 21: Бесконечный скрипт по наполнению каталога

Результат работы:

```
...
New directory size is 2310144, files inside: 106473
New directory size is 2314240, files inside: 106631
New directory size is 2318336, files inside: 107376
New directory size is 2322432, files inside: 107379
New directory size is 2326528, files inside: 107485
New directory size is 2330624, files inside: 107496
```

Листинг 22: Отрывок вывода по бесконечному наполнению

Был приведен отрывок, так как достичь максимального значения врятли бы удалось, так как максимальное количество файлов/каталогов зависит от inode, количество которых варьируется, но максимальное равно $2^{32} - 1$

8 Ознакомиться с содержимым /etc/passwd, /etc/shadow, с утилитой /usr/bin/passwd, проанализировать права доступа к этим файлам.

Файл /etc/passwd содержит строки следующего вида, разделенные двоеточием:
login:password:UID:GID:GECOS:home:shell, где:

- login - имя пользователя
- password - хеш пароляы
- UID - уникальный идентификатор пользователя в пределах системы
- GID - уникальный идентификатор группы в пределах системы, к которой принадлежит пользователь
- GECOS - комментарий, расширенное описание пользователя, например, ФИО
- home - домашний каталог
- shell - начальная оболочка

```
1 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /etc/passwd
2 root:x:0:0:root:/root:/bin/bash
3 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
4 bin:x:2:2:bin:/bin:/usr/sbin/nologin
5 sys:x:3:3:sys:/dev:/usr/sbin/nologin
6 sync:x:4:65534:sync:/bin:/bin/sync
7 ...
```

Листинг 23: Содержимое файла /etc/passwd

В файле /etc/passwd достаточно много пользователей. Данные пользователи необходимы потому, что различные компоненты Linux используют некоторые аккаунты для повышения безопасности. Обычно, такие системные аккаунты имеют идентификатор (uid) меньший 100, и у многих из них в качестве стартовой оболочки установлена /bin/false. Так как эта программа ничего не делает, кроме как выходит и возвращает код ошибки, это эффективно препятствует использованию этих аккаунтов в качестве обычных аккаунтов для логина — т.е. они предназначены только для внутрисистемного пользования.

Файл /etc/shadow в отличии от файла /etc/passwd доступен для чтения только суперпользователю и содержит защифрованную информацию о паролях.

```
1 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo cat /etc/
    ↪ shadow
2 root:*:16951:0:99999:7:::
3 daemon:*:16951:0:99999:7:::
4 bin:*:16951:0:99999:7:::
5 ...
6 psaer:$6$1ikkjGmY$rjvlnuIdtHKYqHZh2r7BwIZyC.
    ↪ NsxIZuh2wkyce6ApBa229J6DIi51o7DA5XkJX0U5vU0cDLtXWfBm/4qF.3F
    ↪ /:17035:0:99999:7:::
7 ...
```

Листинг 24: Содержимое файла /etc/shadow

Каждая строка определяет информацию о пароле конкретного аккаунта, поля в ней разделены знаком ":". Рассмотрим эти поля:

- имя пользователя
- хеш пароля
- дата последнего изменения пароля
- через сколько дней можно будет поменять пароль
- через сколько дней пароль устареет
- за сколько дней до того, как пароль устареет, начать напоминать о необходимости смены пароля
- через сколько дней после того, как пароль устареет, заблокировать учётную запись пользователя
- дата, при достижении которой учётная запись блокируется
- зарезервированное поле

9 Исследовать права владения и доступа, а также их сочетаемость

9.1 Привести примеры применения утилит chmod, chown к специально созданному для этих целей отдельному каталогу с файлами.

Утилита chmod предназначена для изменения прав доступа к файлам и директориям. При добавлении флага -R указанные права доступа также будут применены для вложенных файлов.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder2
2  drwxr-xr-x 2 psaer psaer 4096 окт  2 20:23 folder2
3  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l folder2
4  итого 0
5  -rw-r--r-- 1 psaer psaer 0 окт  2 20:23 temp1
6  -rw-r--r-- 1 psaer psaer 0 окт  2 20:23 temp2
7  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod -R 700
     ↪ folder2
8  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder2
9  drwx------ 2 psaer psaer 4096 окт  2 20:23 folder2
10 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -lR folder2
11 folder2:
12 итого 0
13 -rwx------ 1 psaer psaer 0 окт  2 20:23 temp1
14 -rwx------ 1 psaer psaer 0 окт  2 20:23 temp2
```

Листинг 25: Пример работы chmod

Утилита chown предназначена для изменения владельца и/или группы для указанного файла.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l folder2
итого 0
-rwx------ 1 psaer psaer 0 окт  2 20:23 temp1
-rwx------ 1 psaer psaer 0 окт  2 20:23 temp2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder2
drwx------ 2 psaer psaer 4096 окт  2 20:23 folder2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo chown root:
   ↪ root -R folder2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder2
drwx------ 2 root root 4096 окт  2 20:23 folder2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l folder2
ls: невозможно открыть каталог 'folder2': Отказано в доступе
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo ls -l folder2
итого 0
-rwx------ 1 root root 0 окт  2 20:23 temp1
-rwx------ 1 root root 0 окт  2 20:23 temp2
```

Листинг 26: Пример работы chown

9.2 Расширить права исполнения экспериментального файла с помощью флага SUID.

Обычно в правах доступа присутствуют символы rwx, однако существует символ s. Данный флаг необходим чтобы пользователь смог запустить файл, который ему не принадлежит.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rwxr--r-- 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod u+s temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rwSr--r-- 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l /usr/bin/
   ↪ passwd
-rwsr-xr-x 1 root root 50000 авг 22 15:53 /usr/bin/passwd
```

Листинг 27: Установка SUID

Как видно, наличие флага SUID отображается в списке прав доступа при помощи буквы s вместо x. Если права на выполнение у файла нет, то буква S пишется в верхнем регистре. Как пример утилита **/usr/bin/passwd** имеет данный флаг.

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

Обнулим права пользователя, и предоставим все права группе.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rwSr--r-- 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 077 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
----rwxrwx 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
cat: temp: Отказано в доступе
```

Листинг 28: Формирование прав

Не смотря на то, что пользователь принадлежит к группе, отсутствие у него прав не позволяют получить доступ к файлу. Вывод: приоритет пользовательских прав выше прав группы.

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

Проведем ряд тестов над файлом и каталогом, используя следующий операции: ls, cd, rmdir, cat, cp. Для тестов будем использовать специально созданный каталог и файл:

В первом тесте были даны все возможные права.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 777 folder3
2  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 777 temp
3  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder3
4  drwxrwxrwx 2 psaer psaer 4096 окт  2 21:51 folder3
5  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
6  -rwxrwxrwx 1 psaer psaer 37 окт  2 14:21 temp
7  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cd folder3
8  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings/folder3$ cd ../
9  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ rmdir folder3
10 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
11 1234567890qwertyuiopasdfghjklzxcvbnm
12 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp temp temp2
```

Листинг 29: Тест прав 1

Во втором тесте были даны права лишь на запись.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 222 folder3
2  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 222 temp
3  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder3
4  d-w--w--w- 2 psaer psaer 4096 окт  2 21:53 folder3
5  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
6  --w--w--w- 1 psaer psaer 37 окт  2 14:21 temp
7  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cd folder3
8  bash: cd: folder3: Отказано в доступе
9  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ rmdir folder3
10 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
11 cat: temp: Отказано в доступе
12 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp temp temp2
13 cp: невозможно открыть 'temp' для чтения: Отказано в доступе
```

Листинг 30: Тест прав 2

В третьем тесте были даны права лишь на чтение.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 444 folder3
2  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 444 temp
3  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder3
4  dr--r--r-- 2 psaer psaer 4096 окт  2 21:57 folder3
5  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
6  -r--r--r-- 1 psaer psaer 37 окт  2 14:21 temp
7  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cd folder3
8  bash: cd: folder3: Отказано в доступе
9  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ rmdir folder3
10 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
11 1234567890qwertyuiopasdfghjklzxcvbnm
12 psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp temp temp2
```

Листинг 31: Тест прав 3

В четвертом тесте были убраны все права.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 000 folder3
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ chmod 000 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -ld folder3
d--------- 2 psaer psaer 4096 окт  2 21:58 folder3
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
---------- 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cd folder3
bash: cd: folder3: Отказано в доступе
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ rmdir folder3
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
cat: temp: Отказано в доступе
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cp temp temp2
cp: невозможно открыть 'temp' для чтения: Отказано в доступе
```

Листинг 32: Тест прав 4

10 Разработать «программу-шлюз» для доступа к файлу другого пользователя при отсутствии прав на чтение информации из этого файла. Провести эксперименты для случаев, когда пользователи принадлежат одной и разным группам. Сравнить результаты. Для выполнения задания применить подход, аналогичный для обеспечения функционирования утилиты /usr/bin/passwd (манипуляции с правами доступа, флагом SUID, а также размещением файлов).

В качестве шлюза была написана программа на языке c++.

```
#include <stdio.h>
#include <fstream>
#include <iostream>

using namespace std;

int main(int argc, char ** argv){
        if(argc<2){
                printf("ERROR. No input file!\n");
                return 1;
        }
        ifstream fin(argv[1]);
        if(!fin.is_open()){
                printf("ERROR. Cannot open file");
                return 1;
        }
        char temp;
        while(fin>>temp)
                printf("%c",temp);
        printf("\n");
        fin.close();
        return 0;
}
```

Листинг 33: prog_2_10.cc

Первоначально флаг SUID не установлен. Результаты при работе от владельца:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l prog_2_10
-rwxr-x--- 1 psaer psaer 10416 окт  3 17:52 prog_2_10
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rwx------ 1 psaer psaer 37 окт  2 14:21 temp
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat temp
1234567890qwertyuiopasdfghjklzxcvbnm
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ./prog_2_10 temp
1234567890qwertyuiopasdfghjklzxcvbnm
```

Листинг 34: Запуск от владельца без SUID

```
testuser@psaer-pc:/home/psaer/Study/s7/OS/lab2/otchet/listings$ ./
    ↪ prog_2_10 temp
cat: temp: Отказано в доступе
testuser@psaer-pc:/home/psaer/Study/s7/OS/lab2/otchet/listings$ ./
    ↪ prog_2_10 temp
bash: ./prog_2_10: Отказано в доступе
```

Листинг 35: Запуск от другого пользователя без SUID

Установим флаг SUID.

```
testuser@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l prog_2_10
-rwsr-s--- 1 psaer psaer 10416 окт  3 17:56 prog_2_10
testuser@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ ls -l temp
-rwx------ 1 psaer psaer 37 окт  2 14:21 temp
testuser@psaer-pc:/home/psaer/Study/s7/OS/lab2/otchet/listings$ cat
    ↪ temp
cat: temp: Отказано в доступе
testuser@psaer-pc:/home/psaer/Study/s7/OS/lab2/otchet/listings$ ./
    ↪ prog_2_10 temp
1234567890qwertyuiopasdfghjklzxcvbnm
```

Листинг 36: Запуск от другого пользователя с SUID

Чего и требовалось ожидать, в консоль вывелось содержимое файла.

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

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

Утилита df предоставляет информацию о состоянии жесткого диска и точках монтирования

```
Файловая система Размер Использовано Дост Использовано% Смонтировано в
udev               4,2G            0  4,2G            0% /dev
tmpfs              826M          1,9M  825M            1% /run
/dev/sda2          228G           64G  153G           30% /
tmpfs              4,2G           13M  4,2G            1% /dev/shm
tmpfs              5,3M          4,1k  5,3M            1% /run/lock
tmpfs              4,2G            0  4,2G            0% /sys/fs/cgroup
tmpfs              826M           29k  826M            1% /run/user/1000
tmpfs              826M           50k  826M            1% /run/user/1001
tmpfs              826M           21k  826M            1% /run/user/1002
```

Листинг 37: Утилита df

Как видно, на компьютере 9 точек монтирования.

- udev – менеджер устройств в ядре Linux.
- tmpfs – одна из разновидностей файловых систем, отличающаяся быстрой скоростью работы и надежностью за счет того что располагается в оперативной памяти
- /dev/sda2 - это основной раздел, который отформатирован под Linux

Для каждой файловой системы указан размер, кол-во использованного и доступного пространства. Справа указана точка монтирования. То есть указание, в какую директорию нужно перейти, чтобы оказаться в выбранной файловой системе.

Теперь посмотрим ФС с точки зрения физических устройств, для наглядности были подключены два USB - накопителя

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ lsblk -o +FSTYPE
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT                FSTYPE
sda      8:0    0 223,6G  0 disk
|-sda1   8:1    0     8G  0 part [SWAP]                   swap
|-sda2   8:2    0 215,6G  0 part /                        ext4
sdb      8:16   0 465,7G  0 disk
|-sdb4   8:20   0 465,7G  0 part                         ntfs
sdc      8:32   1   7,5G  0 disk
|-sdc4   8:36   1   7,5G  0 part                         vfat
```

Листинг 38: Утилита lsblk

Как видно в выводе lsblk имеется 3 диска sda, sdb и sdc. Sda является основным жестким диском для моего ноутбука, он разделен на два раздела sda1 и sda2. Sda1 - раздел виртуальной памяти. Sda2 - корневой раздел системы в формате ext4, который характерен для UNIX - подобных систем. Sdb4 имеет тип файловой системы - ntfs который характерен для Windows систем. Sdc4 имеет тип файловой системы -vfat, больше характерен для портативных носителей небольшого объема так как имеет ограничение на максимальный файл в 4 GiB.

Файл /etc/fstab содержит информацию о различных файловых системах и устройствах хранения информации компьютера.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /etc/fstab
# /dev/sda2
UUID=e9d524fd-125e-4108-b897-db016ba19c38       /               ext4
   ↪            rw,relatime,data=ordered        0 1

# /dev/sda1
UUID=ed7cdcfb-5022-49eb-8c62-0397fbf1b70e       none            swap
   ↪            defaults        0 0
```

Листинг 39: Содержимое файла /etc/fstab

Содержимое этого файла имеет следующую структуру:

filesystem	mount point	type	options	dump	pass

filesystem - физическое место размещения файловой системы, по которому определяется

конкретный раздел или устройство хранения для монтирования.

mount point - точка монтирования, куда монтируется корень файловой системы.

type - тип файловой системы. Поддерживается множество типов: ext2, ext3, ext4, btrfs, reiserfs, xfs, jfs, smbfs, iso9660, vfat, ntfs, swap и auto. При выборе auto команда mount попытается определить реальный тип файловой системы самостоятельно.

options - параметры монтирования файловой системы. Рассмотрим встретившиеся параметры:

- **rw** – монтирован только на чтение
- **relatime** – включение записи информации о последнем времени доступа при чтении файла, если предыдущее время доступа (atime) меньше времени изменения файла
- **data=ordered** – журналирование только метаданных
- **defaults** – параметры по умолчанию

dump используется утилитой dump для определения того, нужно ли создать резервную копию данных в файловой системе. Возможные значения: 0 или 1. Если указано число 1, dump создаст резервную копию. У большинства пользователей утилита dump не установлена, поэтому им следует указывать 0 в этом поле.

pass используется программой fsck для определения того, нужно ли проверять целостность файловой системы. Возможные значения: 0, 1 или 2. Значение 1 следует указывать только для корневой файловой системы (с точкой монтирования /); для остальных ФС, которые нужно проверять, используется значение 2, которое имеет менее высокий приоритет. Обратим внимание, что в случае btrfs следует всегда указывать 0, даже если эта файловая система используется в качестве корневой(исключение). Файловые системы, для которых в поле указано значение 0, не будут проверяться fsck.

В файле /etc/mtab прописаны устройства, смонтированые в систему в настоящий момент. При монтировании новой ФС в файл будет добавлена соответствующая запись, а при удалении будет удалена. Формат файла аналогичен формату файла /etc/fstab, за исключение того, что в /etc/fstab физическое место размещения файловой системы показано инодом, а здесь указан путь.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /etc/mtab
2  sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
3  proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
4  udev /dev devtmpfs rw,nosuid,relatime,size=4012084k,nr_inodes=1003021,
     ↪ mode=755 0 0
5  devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,
     ↪ ptmxmode=000 0 0
6  tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=806536k,mode=755 0 0
7  /dev/sda2 / ext4 rw,relatime,data=ordered 0 0
8  securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,
     ↪ relatime 0 0
9  tmpfs /dev/shm tmpfs rw,nosuid,nodev 0 0
10 tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k 0 0
11 tmpfs /sys/fs/cgroup tmpfs ro,nosuid,nodev,noexec,mode=755 0 0
12 cgroup /sys/fs/cgroup/systemd cgroup rw,nosuid,nodev,noexec,relatime,
     ↪ xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=
     ↪ systemd 0 0
13 pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0
```

```
14 cgroup /sys/fs/cgroup/pids cgroup rw,nosuid,nodev,noexec,relatime,pids
   ↪  0 0
15 cgroup /sys/fs/cgroup/devices cgroup rw,nosuid,nodev,noexec,relatime,
   ↪ devices 0 0
16 cgroup /sys/fs/cgroup/cpu,cpuacct cgroup rw,nosuid,nodev,noexec,
   ↪ relatime,cpu,cpuacct 0 0
17 cgroup /sys/fs/cgroup/freezer cgroup rw,nosuid,nodev,noexec,relatime,
   ↪ freezer 0 0
18 cgroup /sys/fs/cgroup/net_cls,net_prio cgroup rw,nosuid,nodev,noexec,
   ↪ relatime,net_cls,net_prio 0 0
19 cgroup /sys/fs/cgroup/perf_event cgroup rw,nosuid,nodev,noexec,
   ↪ relatime,perf_event 0 0
20 cgroup /sys/fs/cgroup/blkio cgroup rw,nosuid,nodev,noexec,relatime,
   ↪ blkio 0 0
21 cgroup /sys/fs/cgroup/cpuset cgroup rw,nosuid,nodev,noexec,relatime,
   ↪ cpuset 0 0
22 systemd-1 /proc/sys/fs/binfmt_misc autofs rw,relatime,fd=32,pgrp=1,
   ↪ timeout=0,minproto=5,maxproto=5,direct 0 0
23 mqueue /dev/mqueue mqueue rw,relatime 0 0
24 debugfs /sys/kernel/debug debugfs rw,relatime 0 0
25 hugetlbfs /dev/hugepages hugetlbfs rw,relatime 0 0
26 binfmt_misc /proc/sys/fs/binfmt_misc binfmt_misc rw,relatime 0 0
27 tmpfs /run/user/1000 tmpfs rw,nosuid,nodev,relatime,size=806536k,mode
   ↪ =700,uid=1000,gid=1000 0 0
28 gvfsd-fuse /run/user/1000/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,
   ↪ relatime,user_id=1000,group_id=1000 0 0
29 fusectl /sys/fs/fuse/connections fusectl rw,relatime 0 0
30 tmpfs /run/user/1001 tmpfs rw,nosuid,nodev,relatime,size=806536k,mode
   ↪ =700,uid=1001,gid=1001 0 0
31 gvfsd-fuse /run/user/1001/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,
   ↪ relatime,user_id=1001,group_id=1001 0 0
32 tmpfs /run/user/1002 tmpfs rw,nosuid,nodev,relatime,size=806536k,mode
   ↪ =700,uid=1002,gid=100 0 0
33 gvfsd-fuse /run/user/1002/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,
   ↪ relatime,user_id=1002,group_id=100 0 0
```

Листинг 40: Содержимое файла /etc/mstab

Список всех доступных для монтирования ФС можно узнать прочитав файл /proc/filesystems. Наличие записи nodev говорит о том, что это виртуальная файловая система.

```
1  psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /proc/
   ↪ filesystems
2  nodev   sysfs
3  nodev   rootfs
4  nodev   ramfs
5  nodev   bdev
6  nodev   proc
7  nodev   cpuset
8  nodev   cgroup
9  nodev   tmpfs
10 nodev   devtmpfs
11 nodev   debugfs
12 nodev   tracefs
13 nodev   securityfs
```

```
nodev   sockfs
nodev   bpf
nodev   pipefs
nodev   devpts
nodev   hugetlbfs
nodev   pstore
nodev   mqueue
        ext3
        ext2
        ext4
nodev   autofs
nodev   binfmt_misc
        fuseblk
nodev   fuse
nodev   fusectl
        vfat
```

Листинг 41: Содержимое файла /proc/filesystems

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

В MBR под таблицу разделов выделено 64 байта. Каждая запись занимает 16 байт. Таким образом, всего на жестком диске может быть создано не более 4 разделов.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo sfdisk -l |
   ↪ tail -n5
Device     Boot   Start     End Sectors  Size Id Type
/dev/sdb1           2048 1789951 1787904  873M 83 Linux
/dev/sdb2        1789952 3049471 1259520  615M 83 Linux
/dev/sdb3        3049472 4104191 1054720  515M 83 Linux
/dev/sdb4        4104192 5210111 1105920  540M 83 Linux
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo fdisk /dev/sdb

Welcome to fdisk (util-linux 2.28).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): n
To create more partitions, first replace a primary with an extended
   ↪ partition.
```

Листинг 42: Попытка создать новый раздел

Однако, данное ограничение обходится если создать расширенный раздел вместо одного физического раздела, в котором можно прописать несколько логических разделов. Таким образом ограничение в ширину отсутствует.

Пример создания 10 разделов:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo sfdisk -l
...
Disk /dev/sdb: 3,8 GiB, 4016046080 bytes, 7843840 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: 0xa667a489

Device     Boot   Start     End Sectors  Size Id Type
/dev/sdb1          2048 1789951 1787904  873M 83 Linux
/dev/sdb2       1789952 3049471 1259520  615M 83 Linux
/dev/sdb3       3049472 4104191 1054720  515M 83 Linux
/dev/sdb4       4104192 5793791 1689600  825M  5 Extended
/dev/sdb5       4106240 4282367  176128   86M 83 Linux
/dev/sdb6       4284416 4632575  348160  170M 83 Linux
/dev/sdb7       4634624 5113855  479232  234M 83 Linux
/dev/sdb8       5115904 5349375  233472  114M 83 Linux
/dev/sdb9       5351424 5412863   61440   30M 83 Linux
/dev/sdb10      5414912 5552127  137216   67M 83 Linux
```

Листинг 43: 10 разделов

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mkdir /
    ↪ depth_test
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mount /dev/
    ↪ sdb1 /depth_test/
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mkdir /
    ↪ depth_test/d1
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mount /dev/
    ↪ sdb2 /depth_test/d1/
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mkdir /
    ↪ depth_test/d1/d2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mount /dev/
    ↪ sdb3 /depth_test/d1/d2/
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mkdir /
    ↪ depth_test/d1/d2/d3
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ sudo mount /dev/
    ↪ sdb5 /depth_test/d1/d2/d3/
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /etc/mtab |
    ↪ grep sdb
/dev/sdb1 /depth_test ext4 rw,relatime,data=ordered 0 0
/dev/sdb2 /depth_test/d1 ext4 rw,relatime,data=ordered 0 0
/dev/sdb3 /depth_test/d1/d2 ext4 rw,relatime,data=ordered 0 0
/dev/sdb5 /depth_test/d1/d2/d3 ext4 rw,relatime,data=ordered 0 0
```

Листинг 44: Монтирование ФС

Из листинга выше видно, что в корневую ФС было успешно смонтировано еще 4 ФС. Таким образом с помощью файла mtab можно построить дерево ФС.

12 Проанализировать и пояснить принцип работы утилиты file.

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.

Утилита идентификации файлов. file выполняет ряд проверок для указанного файла, пытаясь классифицировать его. Сначала происходит тест на файловую систему, затем на магические цифры и затем языковые тесты. Первый пройденный тест прерывает проверку и возвращает результат. Если файл является файлом-директорией текстовым, файлом исходного кода, скомпилированным файлом и так далее, то утилита сообщит об этом.

Тест на магические числа выполняется исходя и информации, содержащейся в файле /usr/share/misc/magic , /etc/magic или /usr/lib/magic (в зависимости от дистрибутива Linux/UNIX).

Рассмотрим заголовок файла prog_2_10 (исполняемый файл, который был создан ранее).

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ od prog_2_10 -cDN25
0000000 177   E   L   F 002 001 001  \0  \0  \0  \0  \0  \0  \0  \0
   ↪ \0
              1179403647            65794                0
   ↪ 0
0000020 002  \0   >  \0 001  \0  \0  \0   0
                 4063234                1               48
0000031
```

Листинг 45: Заголовок файла prog_2_10

Найдем указанное слово ELF в файле /usr/share/mime/magic. Ему соответсвует следующий фрагмент:

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat /usr/share/mime
   ↪ /magic
...
>0=#[]ELF
1>5=##
2>16=##
>0=#[]ELF
...
```

Листинг 46: Частичное содержимое /usr/share/mime/magic

И попробуем проанализировать файл при помощи утилиты file

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file prog_2_10
prog_2_10: setuid, setgid ELF 64-bit LSB executable, x86-64, version 1
   ↪  (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.
   ↪ so.2, for GNU/Linux 2.6.32, BuildID[sha1]=5
   ↪ f98084bf002d2531365b3eadc5d9366d3fcfaf7, not stripped
```

Листинг 47: Использование file

Так как эти строчки совпали, вывелось ранее предопределенное сообщение.

12.2 Утилиту file выполнить с разными ключами.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file -v prog_2_10
file-5.25
magic file from /etc/magic:/usr/share/misc/magic
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file -l temp
#Выведет лист с возможными обнаружениями.
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ touch temp2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file temp2
temp2: empty
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ echo "test">temp2
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file temp2
temp2: ASCII text
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file prog_2_10.cc
prog_2_10.cc: C source, ASCII text
```

Листинг 48: Использование file с различными ключами и файлами

12.3 Привести экспериментальную попытку с добавлением в базу собственного типа файла и его дальнейшей идентификацией. Описать эксперимент и привести последовательность действий для расширения функциональности утилиты file и возможности встраивания дополнительного типа файла в ФС (согласовать содержимое информационной базы и заголовка файла нового типа).

Добавим в etc/magic собственный тип файла с его идентификацией:

```
psaer@psaer-pc:/etc$ sudo gedit magic
```

Листинг 49: Добавление собственного типа

После чего откроется файл, и можно будет дописать нужную строку:

0 string new_file new_file_type

Данная строка означает: искать с 0 бита, искать строку, значение должно быть равно new_file, тип файла определить как new_file_type. Создадим тестовый файл и добавим туда строку new_file. Затем попробуем определить наш тип командой file.

```
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ cat tt
new_file
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file tt
tt: new_file_type
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ rm tt
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ touch tt
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ echo "123">tt
psaer@psaer-pc:~/Study/s7/OS/lab2/otchet/listings$ file tt
tt: ASCII text
```

Листинг 50: Добавление собственного типа

Как можно судить из вывода, определение типа работает корректно. После изменения файла маска не совпала с введенным тестовым типом, поэтому утилита file распознала файл как просто текст в кодировке ASCII.

13 Вывод

В ОС Linux существует несколько типов файлов: обычные, директории, ссылки, сокеты, очереди, блок-ориентированные файлы, байт-ориентированные файлы. Благодаря использовнию

ссылок у одного файла может быть несколько путей, т.е. несколько файлов в структуре каталогов Linux могут быть физически одним файлом на диске. Это достигается тем, что в файловой системе каждый файл идентифицируется уникальным номером, называемым inode (Индексный дескриптор). Индексные дескрипторы хранят информацию о файлах, такую как принадлежность владельцу (пользователю и группе), режим доступа (чтение, запись, запуск на выполнение) и тип файла.

Были разобраны специальные файлы /etc/passwd, /etc/shadow в которых хранятся аатрибуты пользователей.

Познакомился с утилитами:

- file - для определения типа файла
- df/lsblk/sfdisk - инструментами позволяющими получить информацию о используемых дисках
- od - для проведения анализа содержимого файла

14 Список литературы

1. Типы файлов в Unix. URL: http://younglinux.info/filestype
2. Изучение awk. URL: http://rus-linux.net/MyLDP/consol/awk.html
3. Команда hexdump. URL: https://brendanzagaeski.appspot.com/0006.html
4. Команда find. URL: http://linuxguru.ru/good-notes/utilita-find-dlya-poiska-fajlov/
5. Описание /etc/shadow. URL: http://rus-linux.net/MyLDP/BOOKS/lame-10/lame-10/x822.htm
6. Описание /etc/passwd. URL: https://ru.wikipedia.org/wiki//etc/passwd
7. Описание file. URL: https://en.wikipedia.org/wiki/File(command)
8. Описание полей файла /etc/fstab.URL: http://help.ubuntu.ru/wiki/fstab
9. Монтирование систем. URL: https://www.freebsd.org/doc/ru/books/handbook/mount-unmount.html