**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

|  |  |  |
| --- | --- | --- |
| Студент гр. 2303 |  | Карнаухов В.В. |
| Преподаватель |  | Душутина Е.В. |

Санкт-Петербург

2024

**Цель работы.**

**Проанализировать функциональное назначение структурных элементов дерева ФС.**

**Определить размещение корневого каталога (корневой ФС).**

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

Файлы системы могут быть следующих типов:

- Обычный файл

b Специальный файл блочного устройства

c Файл символьного устройства

d Директория

l Символьная ссылка

p FIFO

s Сокет

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.

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.

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

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

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).

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).

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

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.

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

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

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

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).

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.

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.

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

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.

12.2. Утилиту file выполнить с разными ключами.

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)

**Выполнение работы.**

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*

*Файлы системы могут быть следующих типов:*

*- Обычный файл*

*b Специальный файл блочного устройства*

*c Файл символьного устройства*

*d Директория*

*l Символьная ссылка*

*p FIFO*

*s Сокет*

Для поиска образцов каждого типа файла применяется команда *ls -l -R / 2>/dev/null | grep „^x“*, где *x* — произвольный тип файла системы, 2>/dev/null — устранение сообщений об ошибках. Команда выводит только те строки в списке директорий, в которых первым символом является символ на месте x.

Для поиска отдельно взятого типа файла был разработан скрипт *FileSystemObserver.sh*

#!/bin/bash

name=$(realpath "${1:-FileSystemObserver}") # Имя создаваемого файла

rm -f $name

touch -f $name

filetypes="-dcblps"

for (( i=0; i<${#filetypes}; i++ )); do

exp="^${filetypes:$i:1}"

`ls -lR 2>/dev/null / | grep $exp -m 1 -s > $name`

done

Данный скрипт принимает в командной строке имя файла, куда сохраняется вывод.

Результат (сохранен в log.txt):

-rw------- 1 root root 2147483648 апр 17 22:00 swapfile

drwxr-xr-x 4 root root 4096 апр 17 22:05 boot

crw-r--r-- 1 root root 10, 235 апр 23 15:56 autofs

brw-rw---- 1 root disk 7, 0 апр 23 15:56 loop0

lrwxrwxrwx 1 root root 7 апр 17 22:00 bin -> usr/bin

prw------- 1 root root 0 апр 23 15:56 initctl

srw-rw-rw- 1 root root 0 апр 23 15:56 acpid.socket

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.*

Для поиска всех жестких ссылок одного файла был разработан скрипт *HardLinkObserver.sh*:

#!/bin/bash

inode=$(ls -i $1 | cut -d' ' -f1) # Взятие индексного дескриптора у основного файла

ls -l -i -R / 2>/dev/null | grep $inode # Вывод всех жестких ссылок файла

У каждого файла есть свое значение индексного дескриптора, которое можно получить флагом -i. У файлов и их жестких ссылок одинаковые ИД, поэтому скрипт ищет в полном списке дерева такие же значения ИД, как и у исходного файла.

Пример работы (Сохранен в log\_2.txt):

b4ll3r1um@Linex:~/Labs\_OS/LR2/2$ bash HardLinksObserver.sh orig

397153 -rw-rw-r-- 4 b4ll3r1um b4ll3r1um 4 мар 2 18:03 farhardlink

397153 -rw-rw-r-- 4 b4ll3r1um b4ll3r1um 4 мар 2 18:03 hardlink

397153 -rw-rw-r-- 4 b4ll3r1um b4ll3r1um 4 мар 2 18:03 hardlink2

397153 -rw-rw-r-- 4 b4ll3r1um b4ll3r1um 4 мар 2 18:03 orig

b4ll3r1um@Linex:~/Labs\_OS/LR2/2$

Скрипт обнаружил все файлы, значение которых совпало с индексным дескриптором оригинала.

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.*

Создание символьных ссылок через утилиты ln, cp (сохранен в log\_3.txt):

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$ ls -l

total 8

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 6 мар 2 19:31 orig

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 594 мар 5 14:54 SoftLinksObserver.sh

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$ ln -s orig softlink

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$ ls -l

total 8

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 6 мар 2 19:31 orig

lrwxrwxrwx 1 b4ll3r1um b4ll3r1um 4 апр 23 16:28 softlink -> orig

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 594 мар 5 14:54 SoftLinksObserver.sh

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$ cp -P softlink softlinkCopy

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$ ls -l

total 8

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 6 мар 2 19:31 orig

lrwxrwxrwx 1 b4ll3r1um b4ll3r1um 4 апр 23 16:28 softlink -> orig

lrwxrwxrwx 1 b4ll3r1um b4ll3r1um 4 апр 23 16:29 softlinkCopy -> orig

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 594 мар 5 14:54 SoftLinksObserver.sh

b4ll3r1um@Linex:~/Labs\_OS/LR2/3$

Командой ln создала символьную ссылку softlink на файл orig, команда cp создала копию softlinkCopy символьной ссылки softlink. В утилите ls стрелкой указывается на какой файл ссылается ссылка. Первоначальная ссылка и её копия указывают на один и тот же файл.

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

Для поиска всех полноименных символьных ссылок на файл был разработан скрипт SoftLinksObservers.sh:

#!/bin/bash

name="log\_4.txt" # Имя файла

ls -l -R / 2>/dev/null | grep "^l" | grep "$1$" > "$name" # Поиск всех символьных

echo "Total symbolic links: $(wc -l "$name" | cut -d' ' -f1)" >> "$name" # Вывод кол-ва символьных ссылок на данный файл

Вывод (сохранен в log\_4.txt):

lrwxrwxrwx 1 b4ll3r1um b4ll3r1um 4 апр 23 16:44 softlink -> orig

lrwxrwxrwx 1 b4ll3r1um b4ll3r1um 4 апр 23 16:45 softlinkCopy -> orig

Total symbolic links: 2

Скрипт вывел в отдельный файл все найденные символьные ссылки на файл orig, коих всего было найдено 2.

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

Вывод полной информации о каждом файле типа p — FIFO в дереве ФС (сохранен в log\_5.txt):

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ find / -type p -ls 2>/dev/null

84 0 prw-rw-r-- 1 b4ll3r1um b4ll3r1um 0 апр 23 15:56 /run/user/1000/gnome-session-leader-fifo

6 0 p--------- 1 b4ll3r1um b4ll3r1um 0 апр 23 15:56 /run/user/1000/systemd/inaccessible/fifo

334 0 prw------- 1 root root 0 апр 23 15:56 /run/initctl

1564 0 prw------- 1 root root 0 апр 23 15:57 /run/systemd/inhibit/14.ref

1560 0 prw------- 1 root root 0 апр 23 15:57 /run/systemd/inhibit/13.ref

1558 0 prw------- 1 root root 0 апр 23 15:57 /run/systemd/inhibit/12.ref

1510 0 prw------- 1 root root 0 апр 23 15:57 /run/systemd/inhibit/10.ref

1353 0 prw------- 1 root root 0 апр 23 15:56 /run/systemd/inhibit/4.ref

1199 0 prw------- 1 root root 0 апр 23 15:56 /run/systemd/inhibit/3.ref

1197 0 prw------- 1 root root 0 апр 23 15:56 /run/systemd/inhibit/2.ref

1191 0 prw------- 1 root root 0 апр 23 15:56 /run/systemd/inhibit/1.ref

1426 0 prw------- 1 root root 0 апр 23 15:56 /run/systemd/sessions/2.ref

292 0 p--------- 1 root root 0 апр 23 15:56 /run/systemd/inaccessible/fifo

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$

Утилита find обнаружила все файлы типа pipe в ФС.

Создание папки по имени, возвращаемому командой uname:

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ ls -l

total 0

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ mkdir $(uname)

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ ls -l

total 4

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:06 Linux

Создалась директория с именем Linux.

Создание файла по имени, возвращаемому командой logname, и запись туда строки с применением этой же вложенной команды:

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ touch $(logname)

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ ls -l

total 4

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 0 апр 23 17:08 b4ll3r1um

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:06 Linux

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ cat $(logname)

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ echo "This file name is $(logname)" >> $(logname)

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ cat $(logname)

This file name is b4ll3r1um

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$

Добавление в этот же файл вывода команды uname -a:

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ echo $(uname -a) >> $(logname)

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$ cat $(logname)

This file name is b4ll3r1um

Linux Linex 6.5.0-27-generic #28~22.04.1-Ubuntu SMP PREEMPT\_DYNAMIC Fri Mar 15 10:51:06 UTC 2 x86\_64 x86\_64 x86\_64 GNU/Linux

b4ll3r1um@Linex:~/Labs\_OS/LR2/5$

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).*

Применение утилиты od на ранее созданный файл (сохранено в log\_6.txt):

b4ll3r1um@Linex:~/Labs\_OS/LR2/6$ od $(logname)

0000000 064124 071551 063040 066151 020145 060556 062555 064440

0000020 020163 032142 066154 071063 072461 005155 064514 072556

0000040 020170 064514 062556 020170 027066 027065 026460 033462

0000060 063455 067145 071145 061551 021440 034062 031176 027062

0000100 032060 030456 052455 072542 072156 020165 046523 020120

0000120 051120 042505 050115 057524 054504 040516 044515 020103

0000140 071106 020151 060515 020162 032461 030440 035060 030465

0000160 030072 020066 052125 020103 020062 034170 057466 032066

0000200 074040 033070 033137 020064 034170 057466 032066 043440

0000220 052516 046057 067151 074165 000012

0000231

b4ll3r1um@Linex:~/Labs\_OS/LR2/6$ od -c $(logname)

0000000 T h i s f i l e n a m e i

0000020 s b 4 l l 3 r 1 u m \n L i n u

0000040 x L i n e x 6 . 5 . 0 - 2 7

0000060 - g e n e r i c # 2 8 ~ 2 2 .

0000100 0 4 . 1 - U b u n t u S M P

0000120 P R E E M P T \_ D Y N A M I C

0000140 F r i M a r 1 5 1 0 : 5 1

0000160 : 0 6 U T C 2 x 8 6 \_ 6 4

0000200 x 8 6 \_ 6 4 x 8 6 \_ 6 4 G

0000220 N U / L i n u x \n

0000231

b4ll3r1um@Linex:~/Labs\_OS/LR2/6$

В первом случае утилита вывела восьмеричное представление текстового файла. Во втором флаг -c позволила увидеть файл в изначальных символах

Утилита dump позволяет создать резервную копию данных:

b4ll3r1um@Linex:~/Labs\_OS/LR2/6$ sudo dump -0u /

sudo: /etc/sudoers: Not a directory

DUMP: Date of this level 0 dump: Tue Apr 23 17:29:54 2024

DUMP: Dumping /dev/sda3 (/) to /dev/tape

DUMP: Label: none

DUMP: Writing 10 Kilobyte records

DUMP: mapping (Pass I) [regular files]

DUMP: mapping (Pass II) [directories]

DUMP: estimated 12356378 blocks.

DUMP: Volume 1 started with block 1 at: Tue Apr 23 17:29:55 2024

DUMP: dumping (Pass III) [directories]

DUMP: dumping (Pass IV) [regular files]

DUMP: End of tape detected

DUMP: Closing /dev/tape

DUMP: Volume 1 completed at: Tue Apr 23 17:30:27 2024

DUMP: Volume 1 5426160 blocks (5298.98MB)

DUMP: Volume 1 took 0:00:32

DUMP: Volume 1 transfer rate: 169566 kB/s

DUMP: Change Volumes: Mount volume #2

DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") yes

DUMP: Volume 2 started with block 5426131 at: Tue Apr 23 17:32:37 2024

DUMP: Volume 2 begins with blocks from inode 682693

DUMP: End of tape detected

DUMP: Closing /dev/tape

DUMP: Volume 2 completed at: Tue Apr 23 17:32:58 2024

DUMP: Volume 2 5426159 blocks (5298.98MB)

DUMP: Volume 2 took 0:00:21

DUMP: Volume 2 transfer rate: 258387 kB/s

DUMP: Change Volumes: Mount volume #3

DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") yes

DUMP: Volume 3 started with block 10852260 at: Tue Apr 23 17:33:01 2024

DUMP: Volume 3 begins with blocks from inode 1207751

DUMP: Closing /dev/tape

DUMP: Volume 3 completed at: Tue Apr 23 17:33:08 2024

DUMP: Volume 3 1510231 blocks (1474.83MB)

DUMP: Volume 3 took 0:00:07

DUMP: Volume 3 transfer rate: 215747 kB/s

DUMP: 12362490 blocks (12072.74MB) on 3 volume(s)

DUMP: finished in 60 seconds, throughput 206041 kBytes/sec

DUMP: Date of this level 0 dump: Tue Apr 23 17:29:54 2024

DUMP: Date this dump completed: Tue Apr 23 17:33:08 2024

DUMP: Average transfer rate: 214566 kB/s

DUMP: DUMP IS DONE

b4ll3r1um@Linex:~/Labs\_OS/LR2/6$

Для создания копии резервных данных потребуются права суперпользователя, поэтому приписывается *sudo*. -0u — флаги для полного бэкапа и обновления файла *var/lib/dumpdates* после выполнения.

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).*

Определить максимальное кол-во записей в каталоге экспериментальным путем является очень затратным по памяти и по времени, т.к. размер 1 созданного скриптом ниже файла ~8 КБ. Для файловой системы ext4 максимальное кол-во элементов в каталоге — 1 миллиард, то есть для этого необходимо затратить минимум 8000 ГБ.

Для данного пункта был разработан скрипт *DirBomb.sh*:

#!/bin/bash

dir=$(realpath "${1:-DirBomb.sh}") # Имя директории

rmdir $dir

mkdir $dir

edge=$2 # Кол-во создаваемых для исследования файлов в аргументах скрипта

echo "Start size: $(du -hsb $dir)" # Вывод изначального размера директории

for (( i=0; i<=$edge; i++ )) # Создание через цикл файлов в кол-ве аргумента $2

do

touch "$dir/file\_$i.txt"

echo $i >> "$dir/file\_$i.txt"

done

echo "Final size: $(du -hsb $dir)" # Вывод конечного размера директории

for (( i=0; i<=$edge; i++ )) # Удаление всех созданных файлов через цикл

do

rm -f "$dir/file\_$i.txt"

done

echo "Script done. Directory size after removing: $(du -hsb $dir))"

Скрипт создает в указанной папке столько файлов, сколько указано в аргументах скрипта. Затем выводится конечный размер директории, а файлы удаляются.

Пример работы:

b4ll3r1um@Linex:~/Labs\_OS/LR2/7$ bash DirBomb.sh test 1000

Start size: 4096 /home/b4ll3r1um/Labs\_OS/LR2/7/test

Final size: 40759 /home/b4ll3r1um/Labs\_OS/LR2/7/test

Script done. Directory size after removing: 36864 /home/b4ll3r1um/Labs\_OS/LR2/7/test)

b4ll3r1um@Linex:~/Labs\_OS/LR2/7$

Скрипт создал 1000 файлов, которые увеличили размер директории до 40759 байт.

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

Применение команды cat /etc/passwd:

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ 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:/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-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin

systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin

messagebus:x:102:105::/nonexistent:/usr/sbin/nologin

systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin

syslog:x:104:111::/home/syslog:/usr/sbin/nologin

\_apt:x:105:65534::/nonexistent:/usr/sbin/nologin

tss:x:106:113:TPM software stack,,,:/var/lib/tpm:/bin/false

uuidd:x:107:116::/run/uuidd:/usr/sbin/nologin

systemd-oom:x:108:117:systemd Userspace OOM Killer,,,:/run/systemd:/usr/sbin/nologin

tcpdump:x:109:118::/nonexistent:/usr/sbin/nologin

avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin

usbmux:x:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin

dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin

kernoops:x:113:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin

avahi:x:114:121:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin

cups-pk-helper:x:115:122:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin

rtkit:x:116:123:RealtimeKit,,,:/proc:/usr/sbin/nologin

whoopsie:x:117:124::/nonexistent:/bin/false

sssd:x:118:125:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin

speech-dispatcher:x:119:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false

fwupd-refresh:x:120:126:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin

nm-openvpn:x:121:127:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin

saned:x:122:129::/var/lib/saned:/usr/sbin/nologin

colord:x:123:130:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin

geoclue:x:124:131::/var/lib/geoclue:/usr/sbin/nologin

pulse:x:125:132:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin

gnome-initial-setup:x:126:65534::/run/gnome-initial-setup/:/bin/false

hplip:x:127:7:HPLIP system user,,,:/run/hplip:/bin/false

gdm:x:128:134:Gnome Display Manager:/var/lib/gdm3:/bin/false

b4ll3r1um:x:1000:1000:b4ll3r1um,,,:/home/b4ll3r1um:/bin/bash

vboxadd:x:999:1::/var/run/vboxadd:/bin/false

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

/etc/passwd является файлом, содержащим в текстовом формате список пользовательских учётных записей. Является первым и основным источником информации о правах пользователя операционной системы.

Каждая строка файла описывает одного пользователя и содержит семь полей, разделённых двоеточиями:

* регистрационное имя или логин;
* хеш пароля;
* идентификатор пользователя;
* идентификатор группы по умолчанию;
* информационное поле GECOS;
* начальный (он же домашний) каталог;
* регистрационная оболочка, или shell.

Права доступа к /etc/passwd:

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ ls -l /etc/passwd

-rw-r--r-- 1 root root 2944 апр 17 22:28 /etc/passwd

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

Данный файл относится к простым файлам. Для владельца файла root доступно чтение и запись (2-3 биты), для группы root доступно только чтение (4-6 биты), для остальных тоже только чтение (7-9 биты).

Применение команды cat /etc/shadow

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ ls -l /etc/passwd

-rw-r--r-- 1 root root 2944 апр 17 22:28 /etc/passwd

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ cat /etc/shadow

cat: /etc/shadow: Permission denied

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ ls -l /etc/shadow

-rw-r----- 1 root shadow 1562 апр 17 22:28 /etc/shadow

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

Команде не удалось прочитать файл. По правам доступа видно, что каких либо прав у любого пользователя нет (кроме root и группы shadow). Поэтому выполним команду с применением прав суперпользователя:

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ sudo cat /etc/shadow

sudo: /etc/sudoers: Not a directory

root:$y$j9T$tTAaSvl9n65tYSlt0WKaM/$eEvqgEEpa482b5D6vkCH4R.TmlwKvt/8SKgdj2f/WgD:19830:0:99999:7:::

daemon:\*:19773:0:99999:7:::

bin:\*:19773:0:99999:7:::

sys:\*:19773:0:99999:7:::

sync:\*:19773:0:99999:7:::

games:\*:19773:0:99999:7:::

man:\*:19773:0:99999:7:::

lp:\*:19773:0:99999:7:::

mail:\*:19773:0:99999:7:::

news:\*:19773:0:99999:7:::

uucp:\*:19773:0:99999:7:::

proxy:\*:19773:0:99999:7:::

www-data:\*:19773:0:99999:7:::

backup:\*:19773:0:99999:7:::

list:\*:19773:0:99999:7:::

irc:\*:19773:0:99999:7:::

gnats:\*:19773:0:99999:7:::

nobody:\*:19773:0:99999:7:::

systemd-network:\*:19773:0:99999:7:::

systemd-resolve:\*:19773:0:99999:7:::

messagebus:\*:19773:0:99999:7:::

systemd-timesync:\*:19773:0:99999:7:::

syslog:\*:19773:0:99999:7:::

\_apt:\*:19773:0:99999:7:::

tss:\*:19773:0:99999:7:::

uuidd:\*:19773:0:99999:7:::

systemd-oom:\*:19773:0:99999:7:::

tcpdump:\*:19773:0:99999:7:::

avahi-autoipd:\*:19773:0:99999:7:::

usbmux:\*:19773:0:99999:7:::

dnsmasq:\*:19773:0:99999:7:::

kernoops:\*:19773:0:99999:7:::

avahi:\*:19773:0:99999:7:::

cups-pk-helper:\*:19773:0:99999:7:::

rtkit:\*:19773:0:99999:7:::

whoopsie:\*:19773:0:99999:7:::

sssd:\*:19773:0:99999:7:::

speech-dispatcher:!:19773:0:99999:7:::

fwupd-refresh:\*:19773:0:99999:7:::

nm-openvpn:\*:19773:0:99999:7:::

saned:\*:19773:0:99999:7:::

colord:\*:19773:0:99999:7:::

geoclue:\*:19773:0:99999:7:::

pulse:\*:19773:0:99999:7:::

gnome-initial-setup:\*:19773:0:99999:7:::

hplip:\*:19773:0:99999:7:::

gdm:\*:19773:0:99999:7:::

b4ll3r1um:$y$j9T$a/sUQkBzqPquNUln8BU6w/$owCzGOJykjnHKMMcUcreBrFeRhNWAIpalPSL3dolPz5:19830:0:99999:7:::

vboxadd:!:19830::::::

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

Команда вывела содержимое файла. В нем хранятся фактические пароли пользователей в зашифрованном формате.

Структура каждой строки, разделенной двоеточием:

* регистрационное имя или логин;
* Зашифрованный пароль
* Кол-во дней, когда пароль был сменен в последний раз
* Число дней до смены пароля
* Число дней, после которых пароль должен быть сменен
* Число дней, в течение которых пользователь получает предупреждение о необходимости сменить пароль
* Число дней после окончания действия пароля, когда пользователю все еще доступен аккаунт.
* Число дней с 01.01.1970, после которых пароль будет заблокирован
* Остальные поля зарезервированы

Утилита /usr/bin/passwd позволяет сменить пароль:

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ /usr/bin/passwd

Changing password for b4ll3r1um.

Current password:

New password:

BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word

New password:

Retype new password:

passwd: password updated successfully

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

Выясним права доступа для утилиты:

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$ ls -l /usr/bin/passwd

-rwsr-xr-x 1 root root 59976 фев 6 15:54 /usr/bin/passwd

b4ll3r1um@Linex:~/Labs\_OS/LR2/8$

Для владельца root доступно чтение и запись, а также сам файл при исполнении будет иметь те же права, что и его владелец. Для группы root доступно чтение и исполнение. Для остальных тоже чтение и исполнение файла.

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.*

Для дальнейшего исследования воспользуемся данной директорией:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l

total 1268

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:29 1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:16 10

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 5 12:23 11

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:29 12

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:38 2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:44 3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:04 4

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:12 5

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 6

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:46 7

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 8

drwxrwxr-x 5 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 1246200 апр 23 17:59 LR2.docx

b4ll3r1um@Linex:~/Labs\_OS/LR2$

В качестве экспериментальной директории будет использована директория 9:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l 9

total 16

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.4

b4ll3r1um@Linex:~/Labs\_OS/LR2$

Поменяем владельца директории на someuser через chown:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ sudo chown someuser 9

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l

total 1216

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:29 1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:16 10

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 5 12:23 11

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:29 12

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:38 2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:44 3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:04 4

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:12 5

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 6

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:46 7

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 8

drwxrwxr-x 6 someuser b4ll3r1um 4096 апр 23 18:00 9

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 1191972 апр 23 18:04 LR2.docx

b4ll3r1um@Linex:~/Labs\_OS/LR2$

Теперь someuser владеет этой папкой, но не владеет файлами в ней. Флаг -R присваивает каждому объекту в этой директории этого владельца через рекурсивный обход:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ sudo chown -R someuser 9

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l 9

total 16

drwxrwxr-x 2 someuser b4ll3r1um 4096 апр 23 18:00 9.1

drwxrwxr-x 2 someuser b4ll3r1um 4096 апр 23 17:59 9.2

drwxrwxr-x 2 someuser b4ll3r1um 4096 апр 23 17:59 9.3

drwxrwxr-x 2 someuser b4ll3r1um 4096 апр 23 18:00 9.4

b4ll3r1um@Linex:~/Labs\_OS/LR2$

Восстановим владельца:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ sudo chown -R b4ll3r1um 9

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l

total 1128

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:29 1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:16 10

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 5 12:23 11

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:29 12

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:38 2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:44 3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:04 4

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:12 5

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 6

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:46 7

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 8

drwxrwxr-x 6 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 1102679 апр 23 18:12 LR2.docx

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l 9

total 16

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.4

b4ll3r1um@Linex:~/Labs\_OS/LR2$

Права доступа для 3 категорий пользователей остаются те же. Попробуем уравнять права для всех через chmod -R o+w 9:

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l

total 1064

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:29 1

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:16 10

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 5 12:23 11

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 мар 18 22:29 12

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:38 2

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 16:44 3

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:04 4

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:12 5

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 6

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:46 7

drwxrwxr-x 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:58 8

drwxrwxrwx 6 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 1040100 апр 23 18:14 LR2.docx

b4ll3r1um@Linex:~/Labs\_OS/LR2$ ls -l 9

total 16

drwxrwxrwx 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.1

drwxrwxrwx 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.2

drwxrwxrwx 2 b4ll3r1um b4ll3r1um 4096 апр 23 17:59 9.3

drwxrwxrwx 2 b4ll3r1um b4ll3r1um 4096 апр 23 18:00 9.4

b4ll3r1um@Linex:~/Labs\_OS/LR2$

Остальным пользователям (не владельцу и не группе) было выдано право на запись директории и всех файлов. Команда также применила рекурсию, то есть команда применилась для данной директории и всем объектам в ней.

9.2. *Расширить права исполнения экспериментального файла с помощью флага SUID.*

Установим расширенное право SUID для файла file:

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$ ls -l file

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 0 апр 23 18:17 file

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$ chmod u+s file

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$ ls -l

total 0

-rwSrw-r-- 1 b4ll3r1um b4ll3r1um 0 апр 23 18:17 file

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$ chmod u+x file

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$ ls -l

total 0

-rwsrw-r-- 1 b4ll3r1um b4ll3r1um 0 апр 23 18:17 file

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.2$

После выполнения команды право установилось, но оно обозначается как S вместо s. Если право SUID было выдано владельцу, но сам он не обладает правом его исполнения, то оно обозначается как заглавная S. Соответствующее право было выдано файлу.

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

Создадим новый файл text и оставим все права только владельцу:

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ touch text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ echo First string >> text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ echo Second STRING >> text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ echo Third one >> text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ cat text

First string

Second STRING

Third one

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ ls -l

total 4

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 37 апр 23 18:22 text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ chmod u+x text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ chmod go-rw text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ ls -l

total 4

-rwx------ 1 b4ll3r1um b4ll3r1um 37 апр 23 18:22 text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$

Теперь оценим возможности:

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ cat text

First string

Second STRING

Third one

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ echo 4th string >> text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ ./text

./text: line 1: First: command not found

./text: line 2: Second: command not found

./text: line 3: Third: command not found

./text: line 4: 4th: command not found

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$

Отказов в доступе не обнаружено. Теперь передадим эти права группе:

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ chmod u-rwx text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ chmod g+rwx

chmod: missing operand after ‘g+rwx’

Try 'chmod --help' for more information.

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ chmod g+rwx text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ ls -l

total 4

----rwx--- 1 b4ll3r1um b4ll3r1um 48 апр 23 18:27 text

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ cat text

cat: text: Permission denied

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ echo String 5 >> text

bash: text: Permission denied

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$ ./text

bash: ./text: Permission denied

b4ll3r1um@Linex:~/Labs\_OS/LR2/9/9.3$

Теперь работа с файлом не удается.

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).*

Для данного пункта был разработан файл *modBridge.c*:

#include <stdio.h>

int main(int argc, char\* argv[]){

char\* fileName = argv[1];

FILE\* fptr;

fptr = fopen(fileName, "r");

if(!fptr){

printf("Couldn't find or open file\n");

return 1;

}

char string[255];

while(fgets(string, 255, fptr)){

printf("%s", string);

}

return 0;

}

Программа открывает указанный в аргументах файл и выводит его содержимое.

Для эксперимента создадим файл ForbiddenForReading.txt без прав на чтение для группы и остальных пользователей и владельцем root:

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 8

-rwx--x--x 1 b4ll3r1um b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ sudo chown someuser ForbiddenForReading.txt

sudo: /etc/sudoers: Not a directory

[sudo] password for b4ll3r1um:

Sorry, try again.

[sudo] password for b4ll3r1um:

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 8

-rwx--x--x 1 someuser b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$

Скомпилируем программу, дадим ему флаг SUID и сменим владельца:

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ gcc -o src modBridge.c

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 24

-rwx--x--x 1 someuser b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

-rwxrwxr-x 1 b4ll3r1um b4ll3r1um 16144 апр 23 18:55 src

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ chmod u+s src

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ sudo chown someuser src

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 24

-rwx--xr-x 1 someuser b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

-rwxrwxr-x 1 someuser b4ll3r1um 16144 апр 23 18:55 src

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ sudo chmod u+s src

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 24

-rwx-wx-wx 1 someuser b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

-rwsrwxr-x 1 someuser b4ll3r1um 16144 апр 23 18:55 src

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$

Теперь сравним чтение файла через команду и через программу:

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ cat ForbiddenForReading.txt

cat: ForbiddenForReading.txt: Permission denied

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ./src ForbiddenForReading.txt

Кто прочитал тот арбуз

Или картошка

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$

Команда вернула отказ в доступе, а программа выполнилась. Попробуем удалить флаг SUID:

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ sudo chmod u-s src

sudo: /etc/sudoers: Not a directory

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ls -l

total 24

-rwx-wx-wx 1 someuser b4ll3r1um 66 мар 4 14:02 ForbiddenForReading.txt

-rw-rw-r-- 1 b4ll3r1um b4ll3r1um 331 мар 4 14:43 modBridge.c

-rwxrwxr-x 1 someuser b4ll3r1um 16144 апр 23 18:55 src

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$ ./src ForbiddenForReading.txt

Couldn't find or open file

b4ll3r1um@Linex:~/Labs\_OS/LR2/10$

Теперь программа не смогла открыть файл, т.к. при запуске она не приобрела права владельца по флагу SUID.

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.*

Утилита df позволяет проверить использование дискового пространства в ОС. Флаги -T и -h выводят тип файловой системы и использованный/доступный объем памяти в единицах измерения памяти:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11$ df -T

Filesystem Type 1K-blocks Used Available Use% Mounted on

tmpfs tmpfs 1092888 1572 1091316 1% /run

/dev/sda3 ext4 25106692 12305080 11500928 52% /

tmpfs tmpfs 5464420 0 5464420 0% /dev/shm

tmpfs tmpfs 5120 4 5116 1% /run/lock

/dev/sda2 vfat 524252 6220 518032 2% /boot/efi

tmpfs tmpfs 1092884 156 1092728 1% /run/user/1000

/dev/sr0 iso9660 52274 52274 0 100% /media/b4ll3r1um/VBox\_GAs\_7.0.16

b4ll3r1um@Linex:~/Labs\_OS/LR2/11$

Существует аналогичная этой утилита lsblk, с флагом -f она выводит информацию о файловой системе:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11$ lsblk -f

NAME FSTYPE FSVER LABEL UUID FSAVAIL FSUSE% MOUNTPOINTS

loop0 squashfs 4.0 0 100% /snap/core22/1122

loop1 squashfs 4.0 0 100% /snap/firefox/3836

loop2 squashfs 4.0 0 100% /snap/bare/5

loop3 squashfs 4.0 0 100% /snap/gnome-42-2204/141

loop4 squashfs 4.0 0 100% /snap/gtk-common-themes/1535

loop5 squashfs 4.0 0 100% /snap/snap-store/959

loop6 squashfs 4.0 0 100% /snap/snapd/20671

loop7 squashfs 4.0 0 100% /snap/snapd-desktop-integration/83

loop8 0 100% /snap/core22/1380

sda

├─sda1

├─sda2 vfat FAT32 C36A-18E1 505,9M 1% /boot/efi

└─sda3 ext4 1.0 a3ff2ece-d5c3-4413-b2e0-38bc4fc46754 11G 49% /var/snap/firefox/common/host-hunspell

/

sr0 iso9660 Joliet Extension VBox\_GAs\_7.0.16 2024-04-12-17-58-47-24 0 100% /media/b4ll3r1um/VBox\_GAs\_7.0.16

b4ll3r1um@Linex:~/Labs\_OS/LR2/11$

FSTYPE — тип файловой системы

FSVER — версия файловой системы

LABEL — метка файловой системы

UUID — универсальный уникальный идентификатор файловой системы устройства

FSAVAIL — доступный для записи объем памяти

FSUSE — процентный объем занимаемой области памяти

MOUNTPOINTS — каталог, в котором смонтировано устройство

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

Исследованные файловые системы:

* Squashfs — сжимающая файловая система для GNU/Linux, предоставляющая доступ к данным в режиме «только для чтения».
* VFAT — это расширение FAT, появившееся в Windows 95. В FAT имена файлов имеют формат 8.3 и состоят только из символов кодировки ASCII.
* ntfs — стандартная файловая система для семейства операционных систем Windows NT фирмы Microsoft.

NTFS поддерживает хранение метаданных. С целью улучшения производительности, надёжности и эффективности использования дискового пространства для хранения информации о файлах в NTFS используются специализированные структуры данных.

ext4 — журналируемая файловая система, используемая преимущественно в операционных системах с ядром Linux, созданная на базе ext3 в 2006 году.

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.*

Выведем таблицу ФС:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$ cat /proc/partitions

major minor #blocks name

7 0 75996 loop0

7 1 273032 loop1

7 2 4 loop2

7 3 508908 loop3

7 4 93888 loop4

7 5 12620 loop5

7 6 41400 loop6

7 7 452 loop7

11 0 52274 sr0

8 0 26214400 sda

8 1 1024 sda1

8 2 525312 sda2

8 3 25686016 sda3

7 8 76020 loop8

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$

Посмотрим существующие монтирования в ФС командой cat /etc/mtab:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$ 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=5426132k,nr\_inodes=1356533,mode=755,inode64 0 0

devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0

tmpfs /run tmpfs rw,nosuid,nodev,noexec,relatime,size=1092888k,mode=755,inode64 0 0

/dev/sda3 / ext4 rw,relatime,errors=remount-ro 0 0

securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0

tmpfs /dev/shm tmpfs rw,nosuid,nodev,inode64 0 0

tmpfs /run/lock tmpfs rw,nosuid,nodev,noexec,relatime,size=5120k,inode64 0 0

cgroup2 /sys/fs/cgroup cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate,memory\_recursiveprot 0 0

pstore /sys/fs/pstore pstore rw,nosuid,nodev,noexec,relatime 0 0

bpf /sys/fs/bpf bpf rw,nosuid,nodev,noexec,relatime,mode=700 0 0

systemd-1 /proc/sys/fs/binfmt\_misc autofs rw,relatime,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe\_ino=20824 0 0

hugetlbfs /dev/hugepages hugetlbfs rw,relatime,pagesize=2M 0 0

mqueue /dev/mqueue mqueue rw,nosuid,nodev,noexec,relatime 0 0

debugfs /sys/kernel/debug debugfs rw,nosuid,nodev,noexec,relatime 0 0

tracefs /sys/kernel/tracing tracefs rw,nosuid,nodev,noexec,relatime 0 0

fusectl /sys/fs/fuse/connections fusectl rw,nosuid,nodev,noexec,relatime 0 0

configfs /sys/kernel/config configfs rw,nosuid,nodev,noexec,relatime 0 0

ramfs /run/credentials/systemd-sysusers.service ramfs ro,nosuid,nodev,noexec,relatime,mode=700 0 0

/dev/loop2 /snap/bare/5 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop0 /snap/core22/1122 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop1 /snap/firefox/3836 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop3 /snap/gnome-42-2204/141 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop4 /snap/gtk-common-themes/1535 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop5 /snap/snap-store/959 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop6 /snap/snapd/20671 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/loop7 /snap/snapd-desktop-integration/83 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

/dev/sda3 /var/snap/firefox/common/host-hunspell ext4 ro,noexec,noatime,errors=remount-ro 0 0

/dev/sda2 /boot/efi vfat rw,relatime,fmask=0077,dmask=0077,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=remount-ro 0 0

binfmt\_misc /proc/sys/fs/binfmt\_misc binfmt\_misc rw,nosuid,nodev,noexec,relatime 0 0

tmpfs /run/snapd/ns tmpfs rw,nosuid,nodev,noexec,relatime,size=1092888k,mode=755,inode64 0 0

tmpfs /run/user/1000 tmpfs rw,nosuid,nodev,relatime,size=1092884k,nr\_inodes=273221,mode=700,uid=1000,gid=1000,inode64 0 0

gvfsd-fuse /run/user/1000/gvfs fuse.gvfsd-fuse rw,nosuid,nodev,relatime,user\_id=1000,group\_id=1000 0 0

portal /run/user/1000/doc fuse.portal rw,nosuid,nodev,relatime,user\_id=1000,group\_id=1000 0 0

/dev/sr0 /media/b4ll3r1um/VBox\_GAs\_7.0.16 iso9660 ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8 0 0

nsfs /run/snapd/ns/firefox.mnt nsfs rw 0 0

/dev/loop8 /snap/core22/1380 squashfs ro,nodev,relatime,errors=continue,threads=single 0 0

nsfs /run/snapd/ns/snapd-desktop-integration.mnt nsfs rw 0 0

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$

Смонтируем командой mount созданный каталог в /dev/sr0 и найдем через grep результат в mtab:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$ sudo mount /dev/sr0 ./

sudo: /etc/sudoers: Not a directory

mount: /home/b4ll3r1um/Labs\_OS/LR2/11/11.2: WARNING: source write-protected, mounted read-only.

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$ cat /etc/mtab | grep "dev/sr0"

/dev/sr0 /media/b4ll3r1um/VBox\_GAs\_7.0.16 iso9660 ro,nosuid,nodev,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8 0 0

/dev/sr0 /home/b4ll3r1um/Labs\_OS/LR2/11/11.2 iso9660 ro,relatime,nojoliet,check=s,map=n,blocksize=2048,uid=1000,gid=1000,dmode=500,fmode=400,iocharset=utf8 0 0

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.2$

Каталог успешно смонтирован.

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

Найдем максимально возможное дерево ФС в файле /usr/include/linux/limits.h:

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.3$ cat /usr/include/linux/limits.h

/\* SPDX-License-Identifier: GPL-2.0 WITH Linux-syscall-note \*/

#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

b4ll3r1um@Linex:~/Labs\_OS/LR2/11/11.3$

В директиве PATH\_MAX указано 4096. Если размер имени файла содержим минимум 2 символа, то 4096/2 = 2048 директорий.

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

*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.*

*Утилиту file выполнить с разными ключами.*

*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)*

*File* 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.

Синтаксис:

file [-bcLnvz] [-f namefile] [-m magicfile] file ...

Опции:

* -b – не выводить имя файла перед его типом;
* -m magicfile – определяет альтернативный файл с магическими числами; • -с – обычно используется для дебага нового файла с магическими числами перед его использованием;
* -f namefile – определить типы файлов, имена которых записаны в файле namefile;
* -L – определять типы файлов, на которые ссылаются заданные символические ссылки, а не типы ссылок;
* -n – выводить имя файла перед его типом (по умолчанию); • -v – вывести версию программы и выйти;
* -z – пытаться смотреть внутри сжатых файлов.

Запуск утилиты с различными ключами:

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ file -b tmp.c

empty

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ file -b tmp.c

C source, ASCII text

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ file -n tmp.c

tmp.c: C source, ASCII text

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ file -c tmp.c

cont offset type opcode mask value desc

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$

Проведем экспериментальную попытку с добавлением в базу собственного типа файла и его дальнейшей идентификацией. Откроем

/etc/magic с помощью редактора gedit и впишем тип файла lbm:

Файл /etc/magic:

# Magic local data for file(1) command.

# Insert here your local magic data. Format is described in magic(5).

0 string something lbm

Затем создадим файл с таким разрешением и проверим тип:

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ touch code.lbm

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ echo something > code.lbm

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ cat code.lbm

something

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$ file code.lbm

code.lbm: lbm

b4ll3r1um@Linex:~/Labs\_OS/LR2/12$

Утилита определила файл как кастомный.

**Вывод**

В ходе данной лабораторной работы было проанализировано функциональное назначение структурных элементов дерева ФС и определено размещение корневого каталога (корневой ФС).

**СПИСОК ИСТОЧНИКОВ**

1) Losst - Linux Open Source Software Technologies: https://losst.pro/

2) http://linux.bolden.ru/

3) https://andreyex.ru/

4) Информация, выводимая командой man для каждой утилиты

5) https://900913.ru/

6) linuxFAQ: https://linux-faq.ru/

7) OpenNet: https://www.opennet.ru/

8) StackOverflow: https://stackoverflow.com/