

Отчёт по лабораторной работе №5

Информационная безопасность

**Дискреционное разграничение прав в Linux. Исследование влияния
дополнительных атрибутов**

Выполнил: Махорин Иван Сергеевич,
НПИбд-02-21, 1032211221

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	7
3.1	Подготовка лабораторного стенда	7
3.2	Создание программы	7
3.3	Исследование Sticky-бита	15
4	Вывод	22
5	Список литературы. Библиография	23

Список иллюстраций

3.1	Проверка наличия в системе компилятора gss, отключение системы запретов до очередной перезагрузки	7
3.2	Создание программы simpleid.c	8
3.3	Компиляция программы и проверка	8
3.4	Выполнение программы	9
3.5	Выполнение команды id и сравнение с предыдущим пунктом . .	9
3.6	Усложнение программы и сохранение её как simpleid2.c	10
3.7	Компиляция и запуск программы	10
3.8	Выполнение команд от имени суперпользователя	11
3.9	Проверка новых атрибутов и смены владельца файла	11
3.10	Запуск simpleid2 и id, сравнение результатов	12
3.11	Повторение действий для SetGID-бита	12
3.12	Создание программы readfile.c	13
3.13	Компиляция программы	13
3.14	Работа с readfile.c и readfile	14
3.15	Проверка	15
3.16	Проверка атрибута Sticky на директории /tmp	15
3.17	Создание файла file01.txt в директории /tmp со словом test	16
3.18	Просмотр атрибутов, разрешение на чтение и запись для категории пользователей «все остальные»	16
3.19	Попытка чтения файла от пользователя guest2	17
3.20	Попытка дозаписи в файла от пользователя guest2	17
3.21	Проверка содержимого файла	17
3.22	Попытка записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию	18
3.23	Проверка содержимого файла	18
3.24	Попытка удалить файл от имени пользователя guest2	19
3.25	Повышение прав до суперпользователя и снятие атрибута t с директории /tmp	19
3.26	Проверка от пользователя guest2, что атрибута t у директории /tmp нет	20
3.27	Повторение предыдущих шагов	20
3.28	Повышение прав до суперпользователя и возвращение атрибута t на директорию /tmp	21

1 Цель работы

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Теоретическое введение

1. Дополнительные атрибуты файлов Linux

В Linux существует три основных вида прав — право на чтение (read), запись (write) и выполнение (execute), а также три категории пользователей, к которым они могут применяться — владелец файла (user), группа владельца (group) и все остальные (others). Но, кроме прав чтения, выполнения и записи, есть еще три дополнительных атрибута [1].

- **Sticky bit**

Используется в основном для каталогов, чтобы защитить в них файлы. В такой каталог может писать любой пользователь. Но, из такой директории пользователь может удалить только те файлы, владельцем которых он является. Примером может служить директория /tmp, в которой запись открыта для всех пользователей, но нежелательно удаление чужих файлов.

- **SUID (Set User ID)**

Атрибут исполняемого файла, позволяющий запустить его с правами владельца. В Linux приложение запускается с правами пользователя, запустившего указанное приложение. Это обеспечивает дополнительную безопасность т.к. процесс с правами пользователя не сможет получить доступ к важным системным файлам, которые принадлежат пользователю root.

- **SGID (Set Group ID)**

Аналогичен `suid`, но относиться к группе. Если установить `sgid` для каталога, то все файлы созданные в нем, при запуске будут принимать идентификатор группы каталога, а не группы владельца, который создал файл в этом каталоге.

- **Обозначение атрибутов `sticky`, `suid`, `sgid`**

Специальные права используются довольно редко, поэтому при выводе программы `ls -l` символ, обозначающий указанные атрибуты, закрывает символ стандартных прав доступа.

Пример:

```
rwsrwsrwt
```

где первая s — это `suid`, вторая s — это `sgid`, а последняя t — это `sticky bit`

В приведенном примере не понятно, `rwt` — это `rw`- или `rwX`? Определить это просто. Если `t` маленькое, значит `x` установлен. Если `T` большое, значит `x` не установлен. То же самое правило распространяется и на `s`.

В числовом эквиваленте данные атрибуты определяются первым символом при четырехзначном обозначении (который часто опускается при назначении прав), например в правах `1777` — символ `1` обозначает `sticky bit`. Остальные атрибуты имеют следующие числовое соответствие:

1 — установлен `sticky bit`

2 — установлен `sgid`

4 — установлен `suid`

2. Компилятор GCC

`GCC` - это свободно доступный оптимизирующий компилятор для языков `C`, `C++`. Собственно программа `gcc` это некоторая надстройка над группой компиляторов, которая способна анализировать имена файлов, передаваемые ей в качестве аргументов, и определять, какие действия необходимо выполнить. Файлы с расширением `.cc` или `.C` рассматриваются, как файлы на языке `C++`, файлы с расширением `.c` как программы на языке `C`, а файлы с расширением `.o` считаются объектными [2].

3 Выполнение лабораторной работы

3.1 Подготовка лабораторного стенда

Убедимся, что в системе установлен компилятор gcc. После чего отключим систему запретов до очередной перезагрузки системы и выполним проверку



```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал
Вт, 17 сентября 11:58
guest@ismakhorin:/home/guest

[root@ismakhorin guest]# gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=
usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --with-sy
tem-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --enable-plug
in --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu
indirect-function --enable-cet --with-tune=generic --with-arch_64=x86_64-v2 --with-arch_32=x86_64 --build=x86_64-redhat-linux --with-build-config=bootstrap-l
o --enable-link-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
[root@ismakhorin guest]# setenforce 0
[root@ismakhorin guest]# getenforce
Permissive
[root@ismakhorin guest]#
```

Рис. 3.1: Проверка наличия в системе компилятора gcc, отключение системы запретов до очередной перезагрузки

3.2 Создание программы

1. Войдём в систему от имени пользователя guest и создадим программу simpleid.c

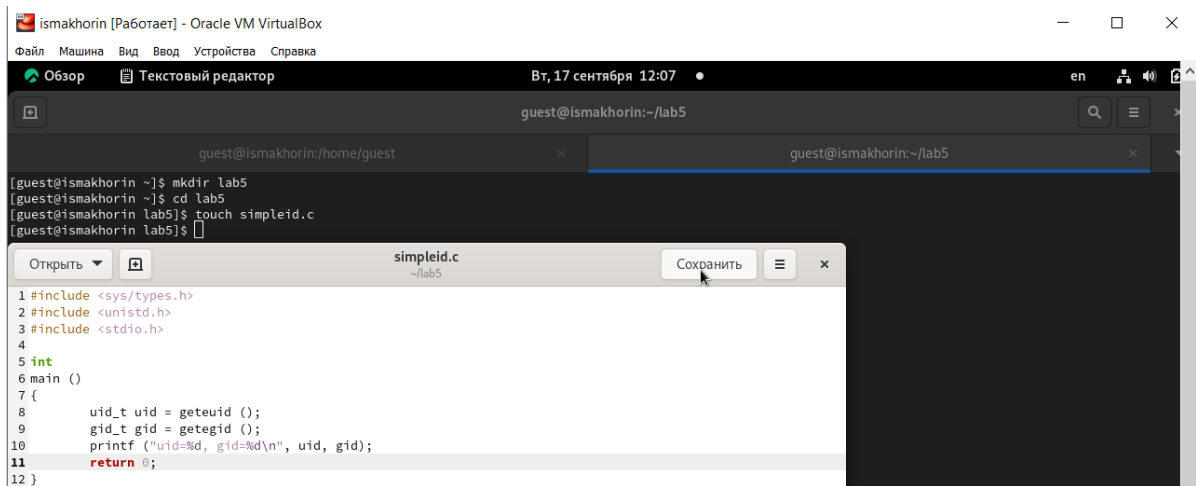


Рис. 3.2: Создание программы simpleid.c

2. Скомпилируем программу и убедимся, что файл программы создан

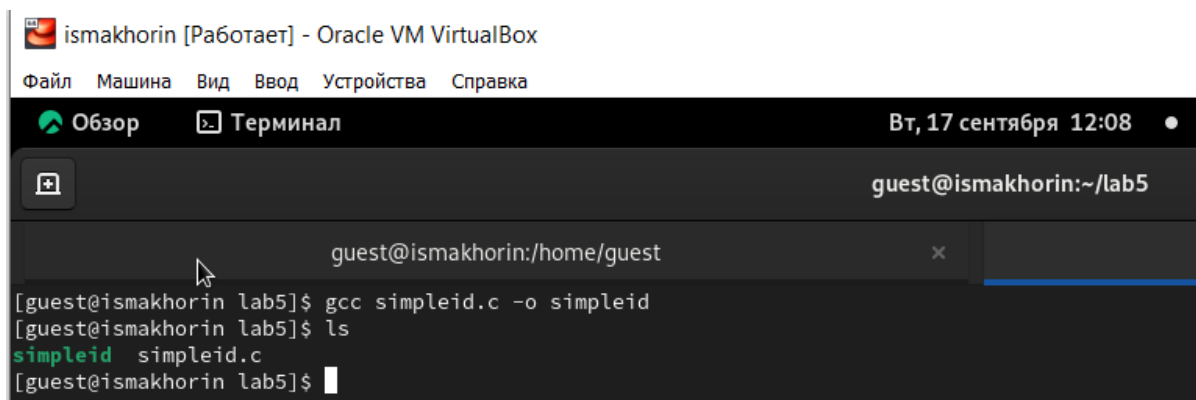


Рис. 3.3: Компиляция программы и проверка

3. Затем выполним программу simpleid

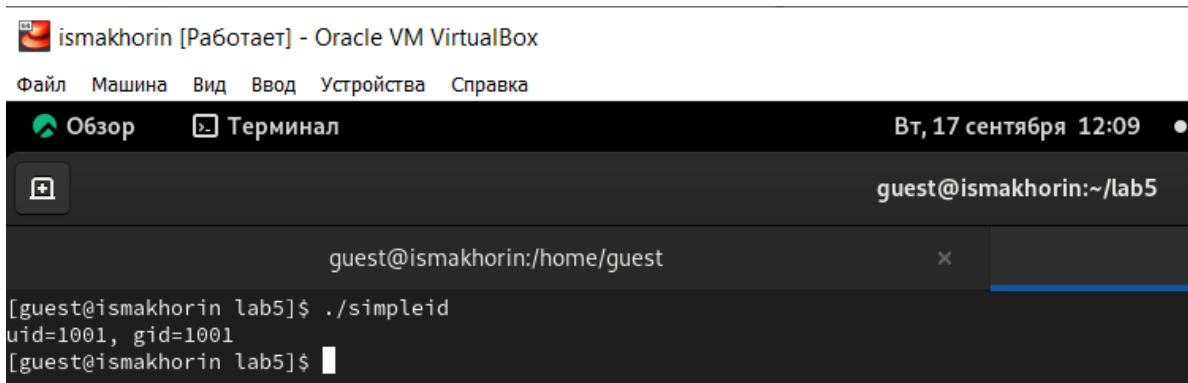


Рис. 3.4: Выполнение программы

4. Выполним системную программу `id` и сравним полученный нами результат с данными предыдущего пункта задания

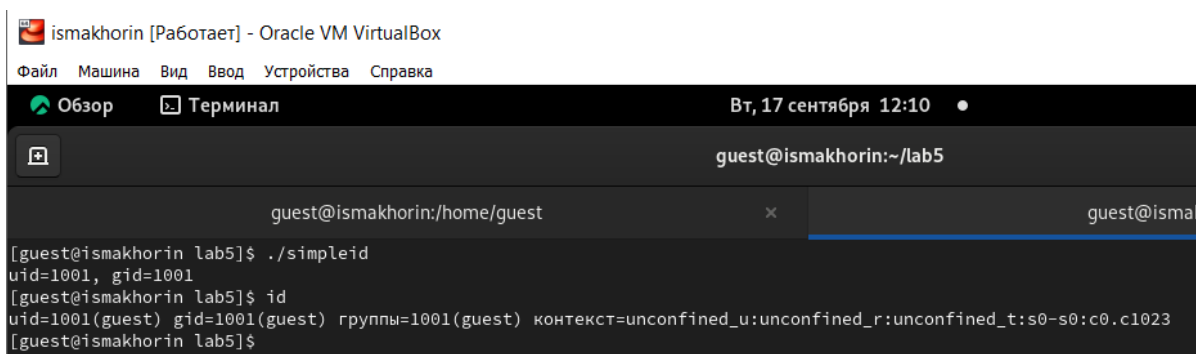
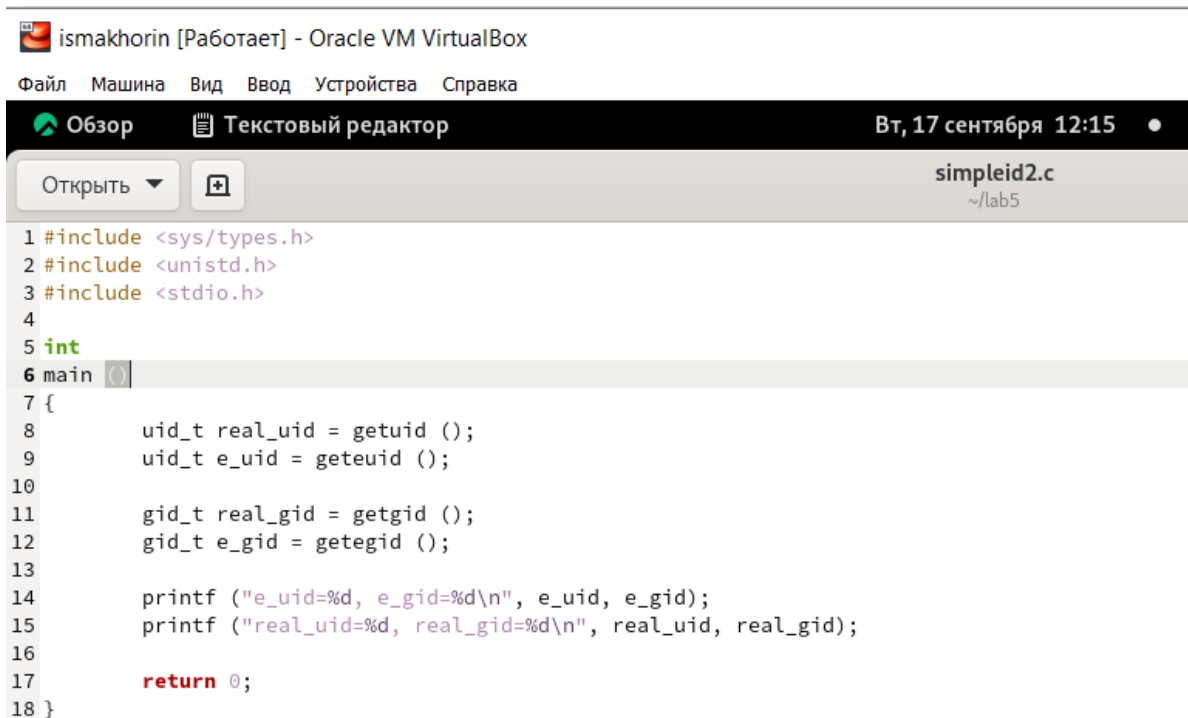


Рис. 3.5: Выполнение команды `id` и сравнение с предыдущим пунктом

5. Усложним программу, добавив вывод действительных идентификаторов. После чего получившуюся программу назовём `simpleid2.c`

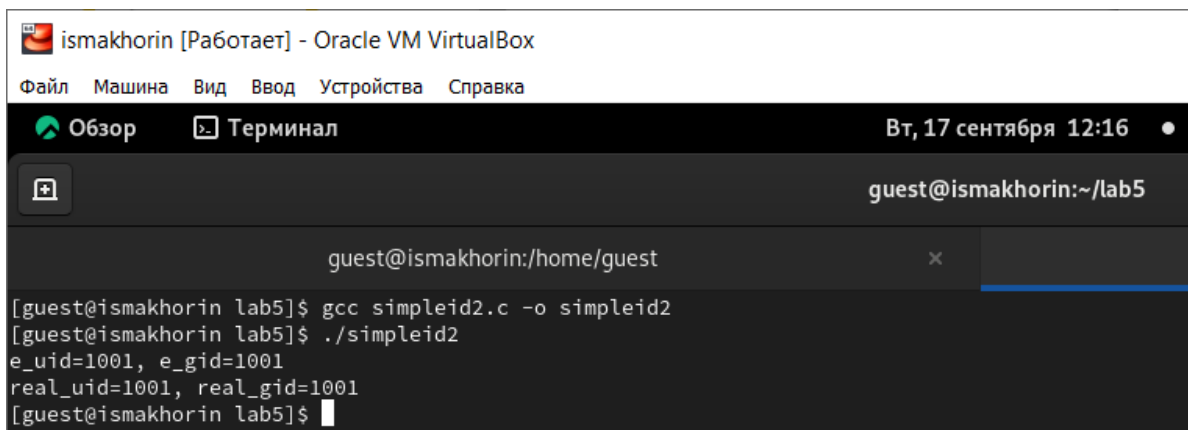


The screenshot shows a text editor window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The toolbar has "Обзор" and "Текстовый редактор" buttons. The file name is "simpleid2.c" located at "~/lab5". The code is as follows:

```
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main ()
7 {
8     uid_t real_uid = getuid ();
9     uid_t e_uid = geteuid ();
10
11     gid_t real_gid = getgid ();
12     gid_t e_gid = getegid ();
13
14     printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
15     printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
16
17     return 0;
18 }
```

Рис. 3.6: Усложнение программы и сохранение её как simpleid2.c

6. Скомпилируем и запустим simpleid2.c



The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The toolbar has "Обзор" and "Терминал" buttons. The prompt is "guest@ismakhorin:~/lab5". The commands and output are as follows:

```
guest@ismakhorin:/home/guest
[guest@ismakhorin lab5]$ gcc simpleid2.c -o simpleid2
[guest@ismakhorin lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ismakhorin lab5]$
```

Рис. 3.7: Компиляция и запуск программы

7. От имени суперпользователя выполним команды

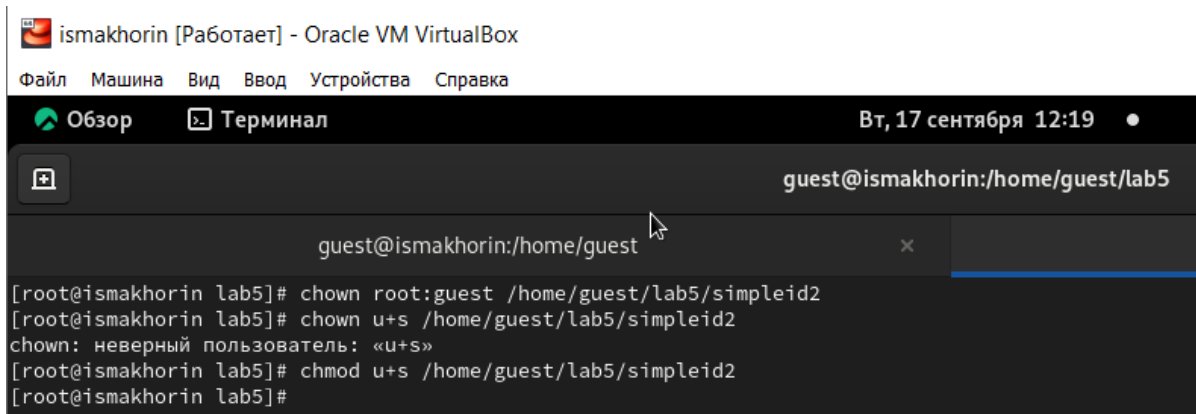


Рис. 3.8: Выполнение команд от имени суперпользователя

8. Повысив временно свои права с помощью `su`, поясним, что делают эти команды

От имени суперпользователя выполнил команды “`sudo chown root:guest /home/guest/simpleid2`” и “`sudo chmod u+s /home/guest/simpleid2`”, затем выполнил проверку правильности установки новых атрибутов и смены владельца файла `simpleid2` командой “`sudo ls -l /home/guest/simpleid2`”. Этими командами была произведена смена пользователя файла на `root` и установлен SetUID-бит.

9. Далее выполним проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`

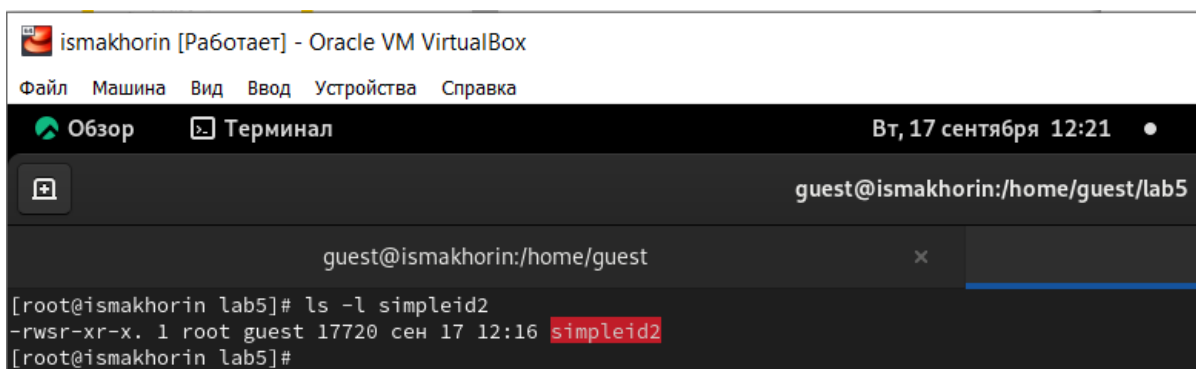
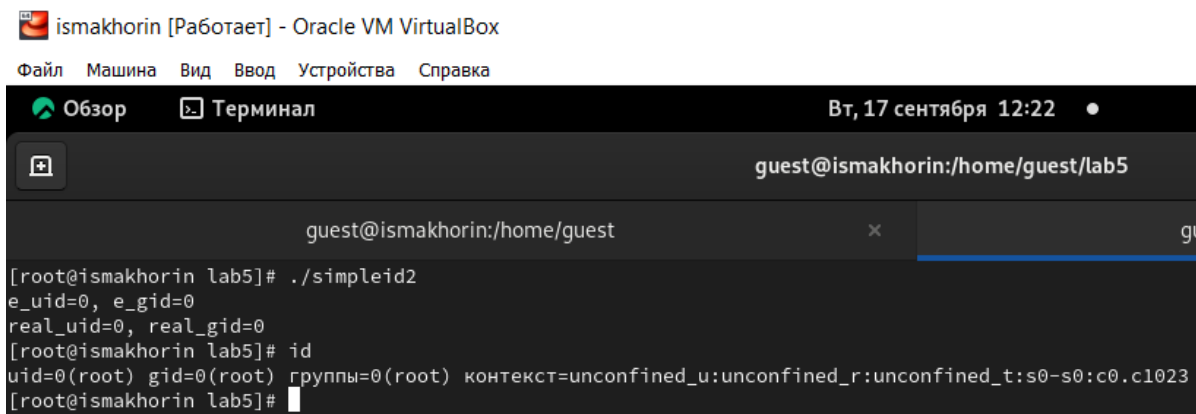


Рис. 3.9: Проверка новых атрибутов и смены владельца файла

10. Запустим `simpleid2` и `id` и сравним результаты

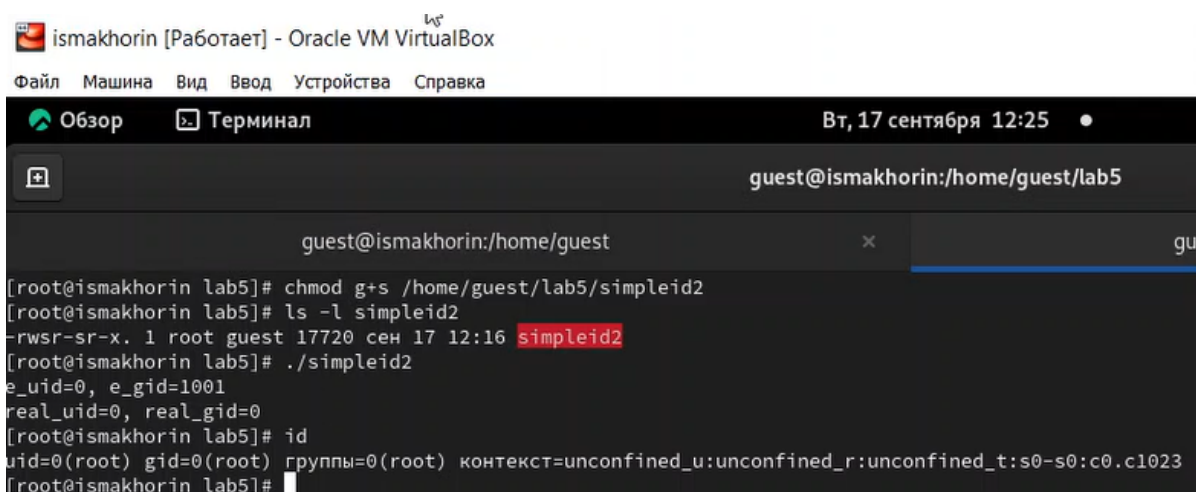


The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The window has tabs for "Обзор" and "Терминал". The terminal prompt is "guest@ismakhorin:/home/guest/lab5". The user runs the command `./simpleid2`, which outputs `e_uid=0, e_gid=0` and `real_uid=0, real_gid=0`. Then, the user runs `id`, which outputs `uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`.

```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал
Вт, 17 сентября 12:22
guest@ismakhorin:/home/guest/lab5
guest@ismakhorin:/home/guest
[root@ismakhorin lab5]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@ismakhorin lab5]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ismakhorin lab5]#
```

Рис. 3.10: Запуск simpleid2 и id, сравнение результатов

11. Теперь проделаем тоже самое относительно SetGID-бита



The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The window has tabs for "Обзор" and "Терминал". The terminal prompt is "guest@ismakhorin:/home/guest/lab5". The user runs `chmod g+s /home/guest/lab5/simpleid2`. Then, the user runs `ls -l simpleid2`, which outputs `-rwsr-sr-x. 1 root guest 17720 сен 17 12:16 simpleid2`. Finally, the user runs `./simpleid2`, which outputs `e_uid=0, e_gid=1001` and `real_uid=0, real_gid=0`. Then, the user runs `id`, which outputs `uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023`.

```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал
Вт, 17 сентября 12:25
guest@ismakhorin:/home/guest/lab5
guest@ismakhorin:/home/guest
[root@ismakhorin lab5]# chmod g+s /home/guest/lab5/simpleid2
[root@ismakhorin lab5]# ls -l simpleid2
-rwsr-sr-x. 1 root guest 17720 сен 17 12:16 simpleid2
[root@ismakhorin lab5]# ./simpleid2
e_uid=0, e_gid=1001
real_uid=0, real_gid=0
[root@ismakhorin lab5]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ismakhorin lab5]#
```

Рис. 3.11: Повторение действий для SetGID-бита

12. Создадим программу readfile.c

ismakhorin [Работает] - Oracle VM VirtualBox

Файл Машина Вид Ввод Устройства Справка

Обзор Текстовый редактор Вт, 17 сентября 12:33

guest@ismakhorin:/home/guest/lab5

```
[root@ismakhorin lab5]# touch readfile.c
[root@ismakhorin lab5]# chmod 777 /home/guest/lab5/readfile.c
[root@ismakhorin lab5]#
```

Открыть *readfile.c ~/lab5

```
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6 int
7 main (int argc, char* argv[])
8 {
9     unsigned char buffer[16];
10    size_t bytes_read;
11    int i;
12    int fd = open (argv[1], O_RDONLY);
13    do
14    {
15        bytes_read = read (fd, buffer, sizeof (buffer));
16        for (i = 0; i < bytes_read; ++i) printf ("%c", buffer[i]);
17    }
18    while (bytes_read == sizeof (buffer));
19    close (fd);
20    return 0;
21 }
```

Рис. 3.12: Создание программы readfile.c

13. Теперь её откомпилируем

ismakhorin [Работает] - Oracle VM VirtualBox

Файл Машина Вид Ввод Устройства Справка

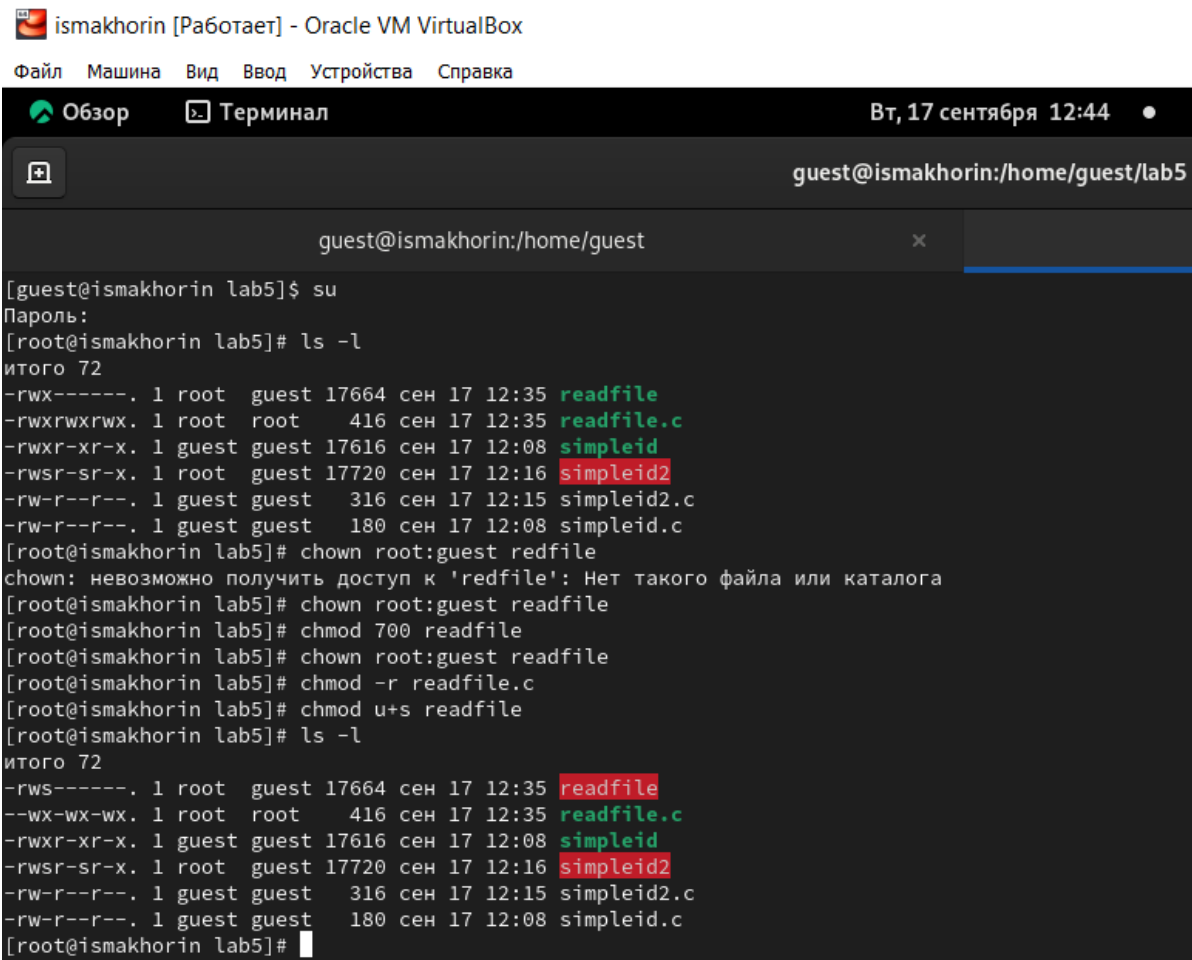
Обзор Терминал Вт, 17 сентября 12:36

guest@ismakhorin:/home/guest/lab5

```
[root@ismakhorin lab5]# gcc readfile.c -o readfile
[root@ismakhorin lab5]#
```

Рис. 3.13: Компиляция программы

14. Сменим владельца у файла readfile.c и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. Затем сменим у программы readfile владельца и установим SetU'D-бит

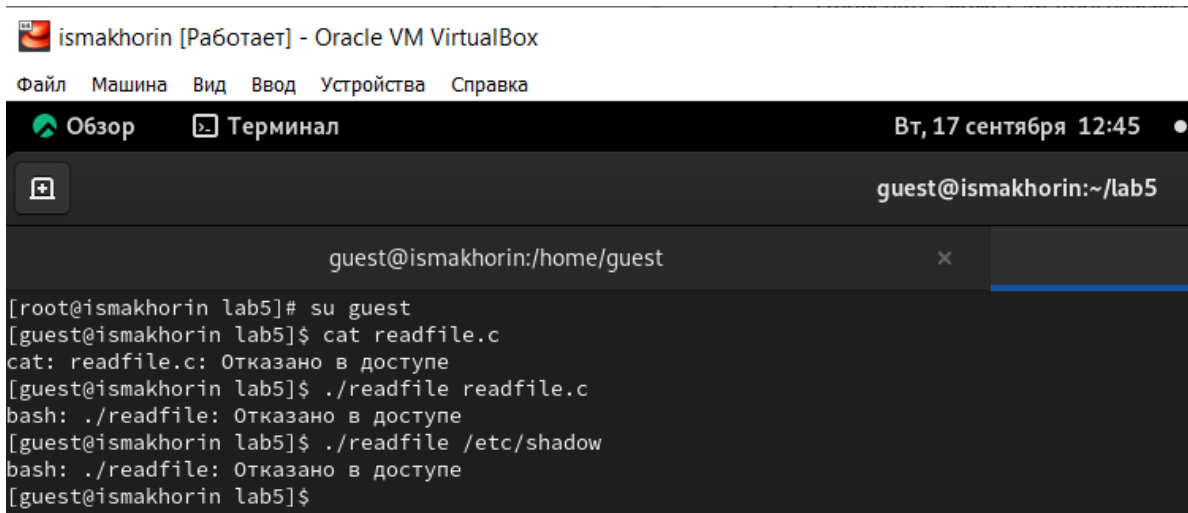


The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The terminal is running a series of commands to change the ownership and permissions of files. The initial state shows files owned by 'root' or 'guest' with various permissions. The user 'root' is used to change the ownership of 'readfile' to 'root:guest' and 'readfile.c' to 'root:guest'. The permissions for 'readfile.c' are changed to '700' and 'u+s' is added to 'readfile'. The final state shows 'readfile' owned by 'root:guest' with permissions 'rws-----' and 'readfile.c' owned by 'root:root' with permissions '--wx-wx-wx'.

```
[guest@ismakhorin lab5]$ su
Пароль:
[root@ismakhorin lab5]# ls -l
итого 72
-rwx-----, 1 root  guest 17664 сен 17 12:35 readfile
-rwxrwxrwx, 1 root  root   416 сен 17 12:35 readfile.c
-rwxr-xr-x, 1 guest guest 17616 сен 17 12:08 simpleid
-rwsr-sr-x, 1 root  guest 17720 сен 17 12:16 simpleid2
-rw-r--r--, 1 guest guest   316 сен 17 12:15 simpleid2.c
-rw-r--r--, 1 guest guest   180 сен 17 12:08 simpleid.c
[root@ismakhorin lab5]# chown root:guest readfile
chown: невозможно получить доступ к 'readfile': Нет такого файла или каталога
[root@ismakhorin lab5]# chown root:guest readfile
[root@ismakhorin lab5]# chmod 700 readfile
[root@ismakhorin lab5]# chown root:guest readfile
[root@ismakhorin lab5]# chmod -r readfile.c
[root@ismakhorin lab5]# chmod u+s readfile
[root@ismakhorin lab5]# ls -l
итого 72
-rws-----, 1 root  guest 17664 сен 17 12:35 readfile
--wx-wx-wx, 1 root  root   416 сен 17 12:35 readfile.c
-rwxr-xr-x, 1 guest guest 17616 сен 17 12:08 simpleid
-rwsr-sr-x, 1 root  guest 17720 сен 17 12:16 simpleid2
-rw-r--r--, 1 guest guest   316 сен 17 12:15 simpleid2.c
-rw-r--r--, 1 guest guest   180 сен 17 12:08 simpleid.c
[root@ismakhorin lab5]#
```

Рис. 3.14: Работа с readfile.c и readfile

15. Проверим, что пользователь guest не может прочитать файл readfile.c, может ли программа readfile прочитать файл readfile.c, и может ли программа readfile прочитать файл /etc/shadow?

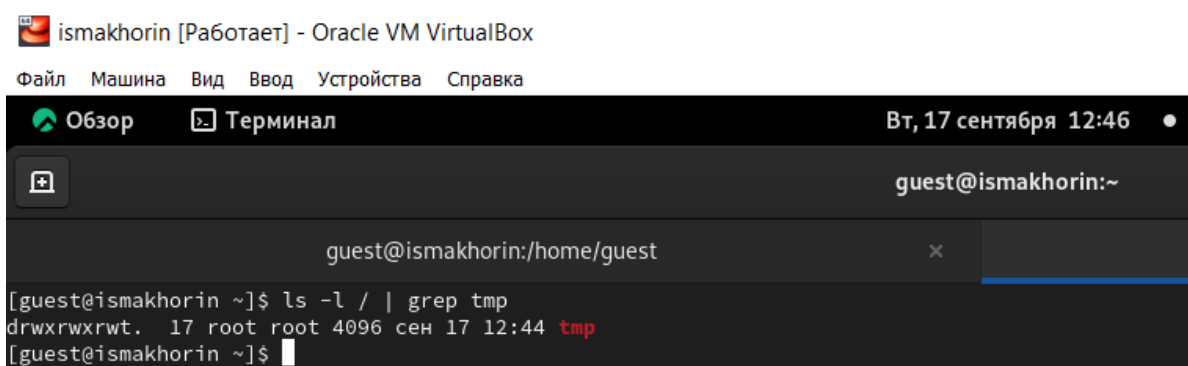


The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The window has tabs for "Обзор" and "Терминал". The terminal shows the user "guest@ismakhorin:~/lab5". The command history is as follows:
[root@ismakhorin lab5]# su guest
[guest@ismakhorin lab5]\$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@ismakhorin lab5]\$./readfile readfile.c
bash: ./readfile: Отказано в доступе
[guest@ismakhorin lab5]\$./readfile /etc/shadow
bash: ./readfile: Отказано в доступе
[guest@ismakhorin lab5]\$

Рис. 3.15: Проверка

3.3 Исследование Sticky-бита

1. Выясним, установлен ли атрибут Sticky на директории /tmp



The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The window has tabs for "Обзор" and "Терминал". The terminal shows the user "guest@ismakhorin:~". The command history is as follows:
[guest@ismakhorin ~]\$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 сен 17 12:44 tmp
[guest@ismakhorin ~]\$

Рис. 3.16: Проверка атрибута Sticky на директории /tmp

2. От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test

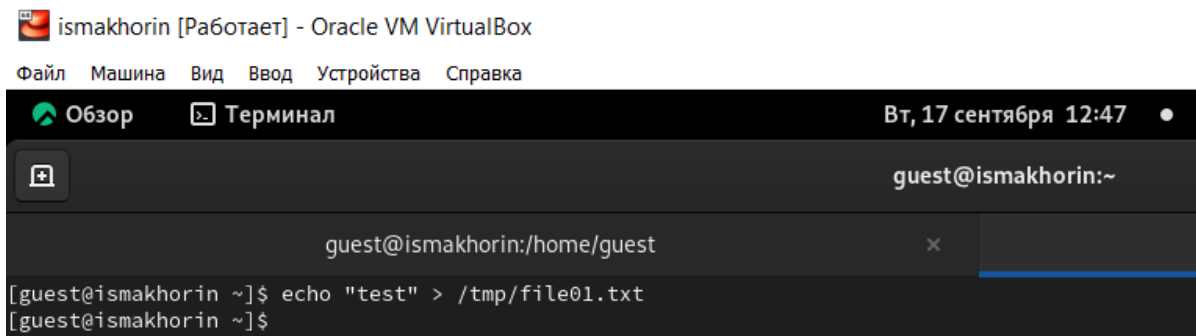


Рис. 3.17: Создание файла file01.txt в директории /tmp со словом test

3. Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные»

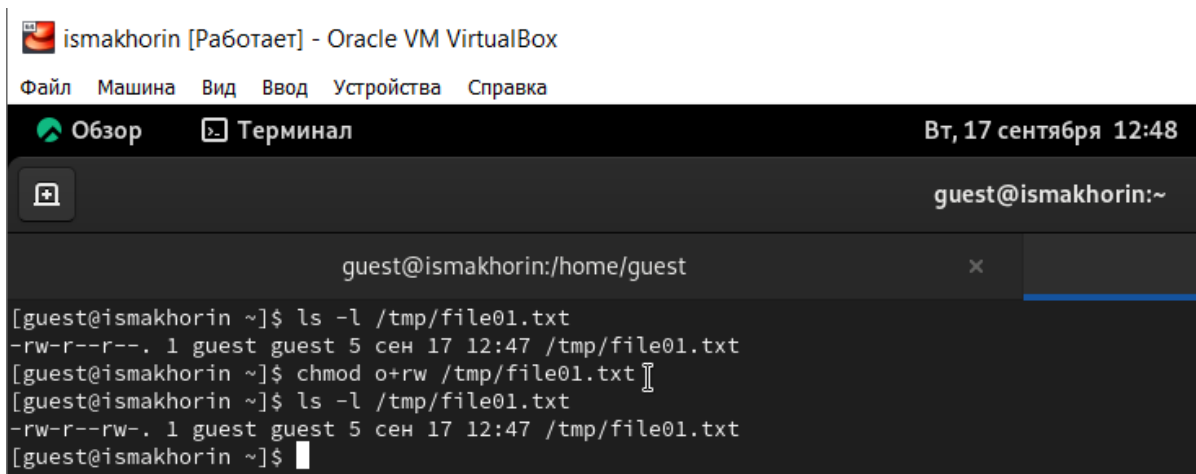


Рис. 3.18: Просмотр атрибутов, разрешение на чтение и запись для категории пользователей «все остальные»

4. От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt

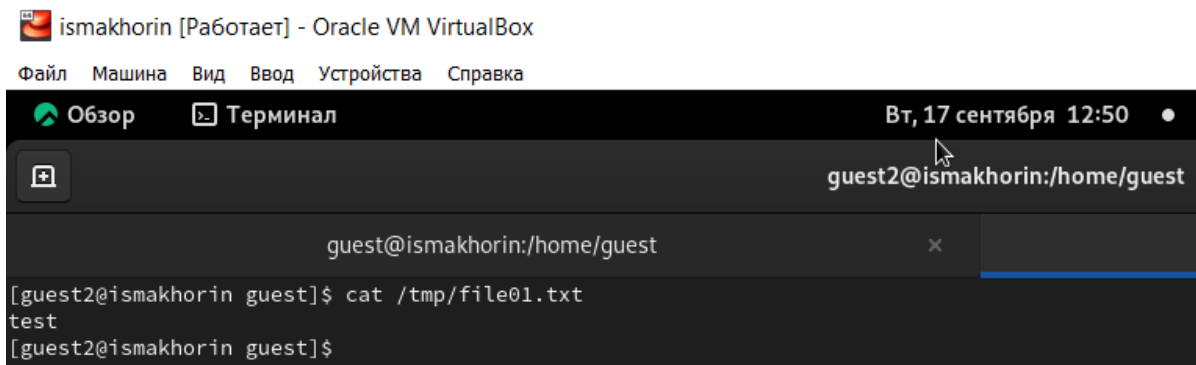


Рис. 3.19: Попытка чтения файла от пользователя guest2

5. От пользователя guest2 попробуем дозаписать в файл

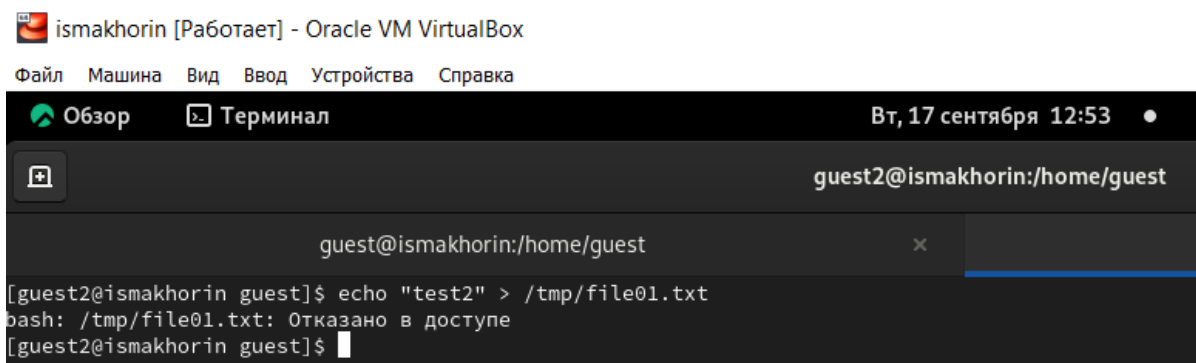


Рис. 3.20: Попытка дозаписи в файла от пользователя guest2

6. Проверим содержимое файла

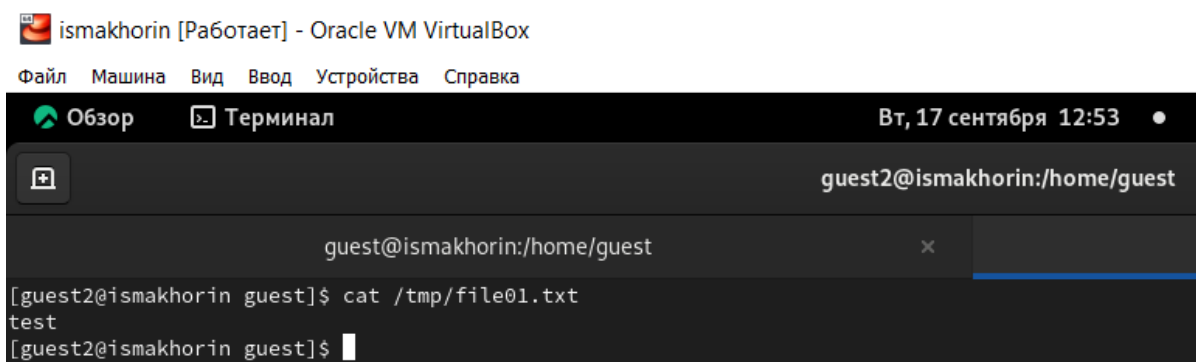
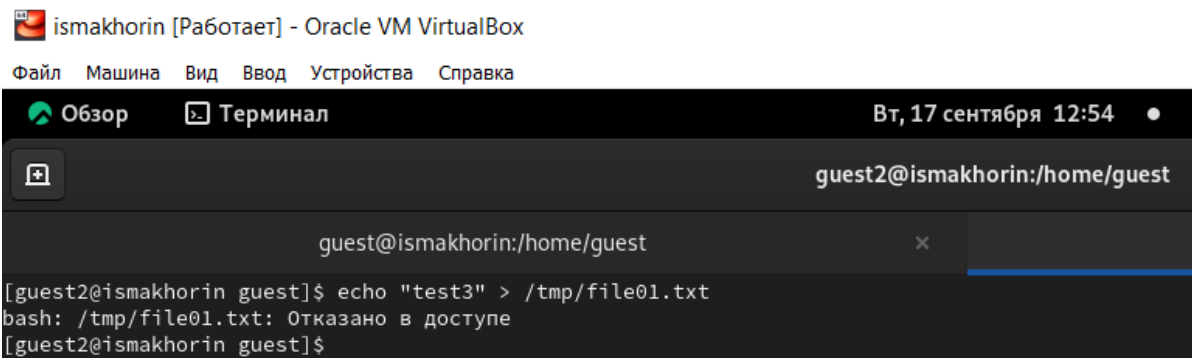


Рис. 3.21: Проверка содержимого файла

7. От пользователя guest2 попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию

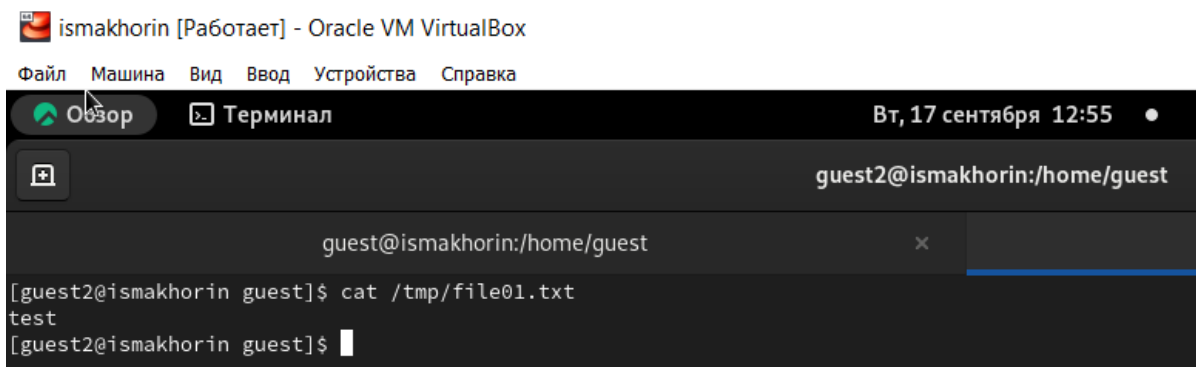


The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The window has two tabs: "Обзор" and "Терминал". The terminal shows the user "guest2@ismakhorin:/home/guest" and the prompt "guest@ismakhorin:/home/guest". The command entered is `echo "test3" > /tmp/file01.txt`, which results in the error message `bash: /tmp/file01.txt: Отказано в доступе`.

```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал  Вт, 17 сентября 12:54
guest2@ismakhorin:/home/guest
guest@ismakhorin:/home/guest
[guest2@ismakhorin guest]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Отказано в доступе
[guest2@ismakhorin guest]$
```

Рис. 3.22: Попытка записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию

8. Проверим содержимое файла



The screenshot shows the same terminal window as before, but now the user "guest2@ismakhorin:/home/guest" has entered the command `cat /tmp/file01.txt`. The output of the command is the word "test".

```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал  Вт, 17 сентября 12:55
guest2@ismakhorin:/home/guest
guest@ismakhorin:/home/guest
[guest2@ismakhorin guest]$ cat /tmp/file01.txt
test
[guest2@ismakhorin guest]$
```

Рис. 3.23: Проверка содержимого файла

9. От пользователя guest2 попробуем удалить файл /tmp/file01.txt

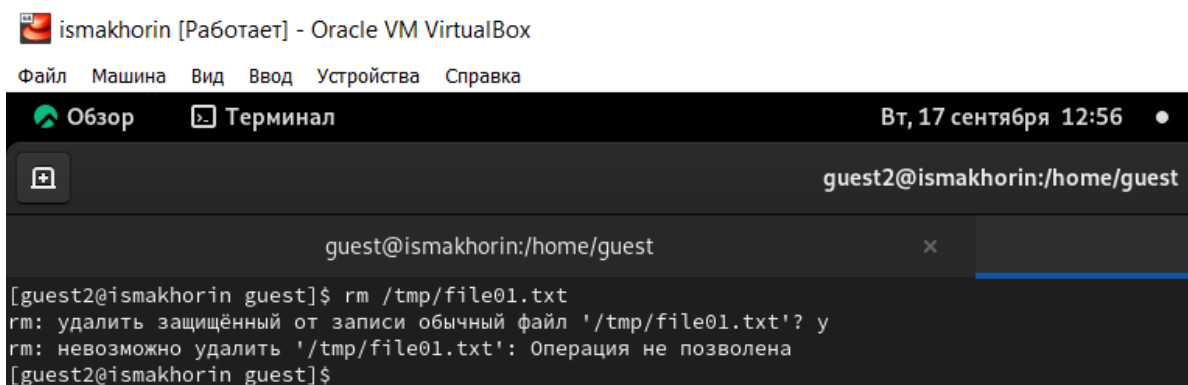


Рис. 3.24: Попытка удалить файл от имени пользователя guest2

10. Повысим свои права до суперпользователя командой `su` и выполним после этого команду, снимающую атрибут `t` (Sticky-бит) с директории `/tmp`

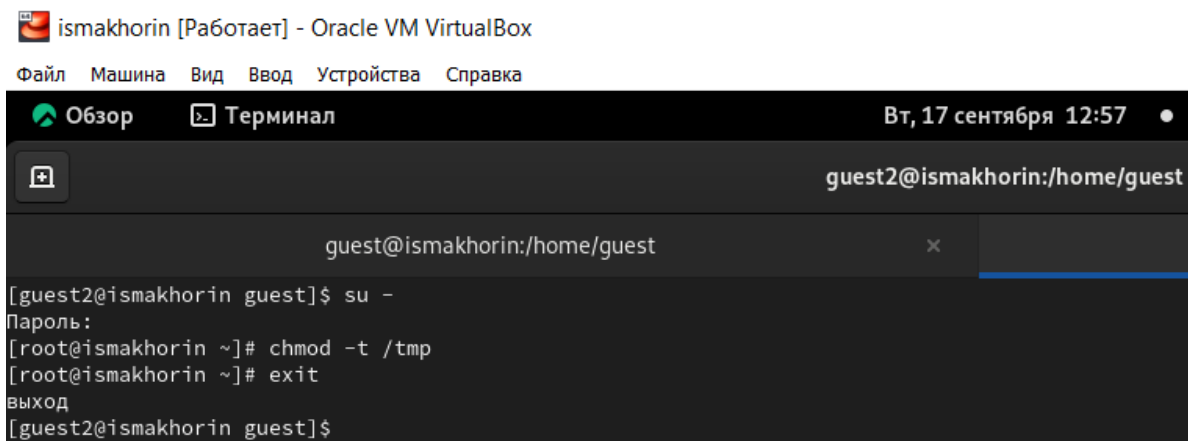


Рис. 3.25: Повышение прав до суперпользователя и снятие атрибута `t` с директории `/tmp`

11. Покинем режим суперпользователя и от пользователя `guest2` проверим, что атрибута `t` у директории `/tmp` нет

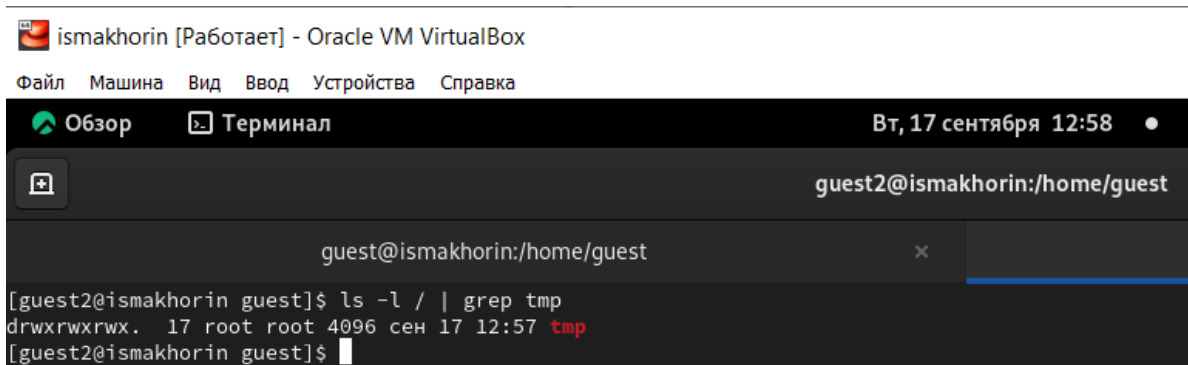


Рис. 3.26: Проверка от пользователя guest2, что атрибута t у директории /tmp нет

12. Повторим предыдущие шаги

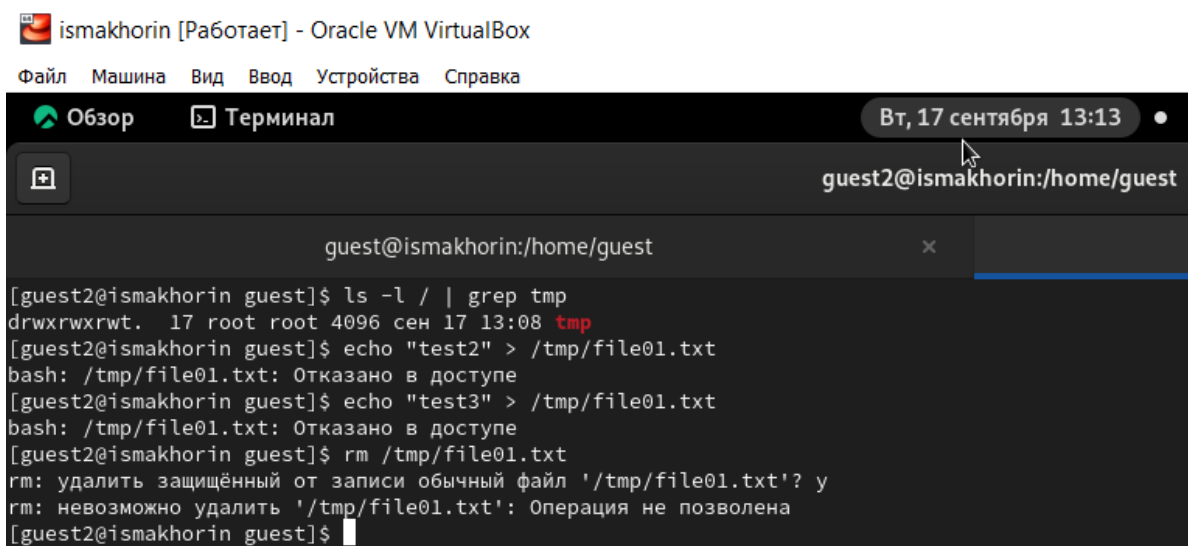
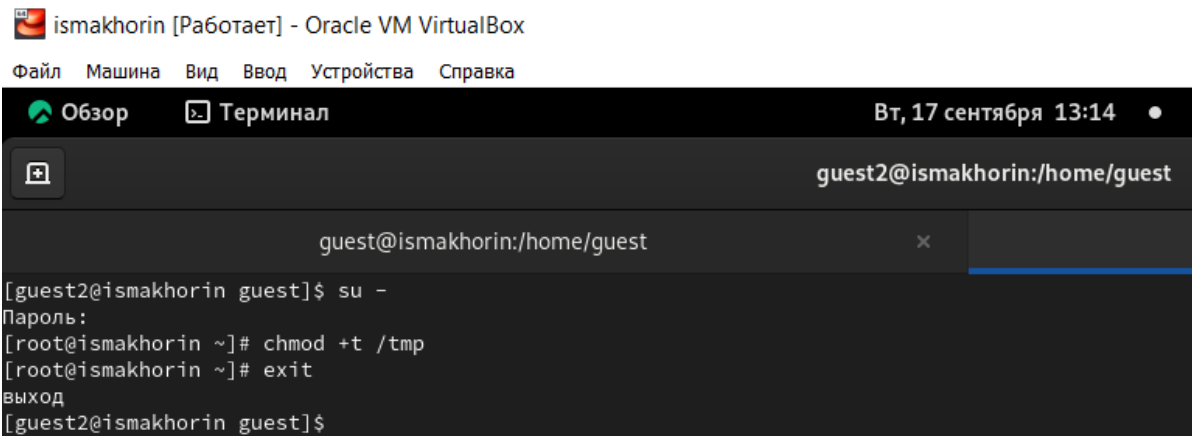


Рис. 3.27: Повторение предыдущих шагов

13. Повысим свои права до суперпользователя и вернём атрибут t на директорию /tmp



The screenshot shows a terminal window titled "ismakhorin [Работает] - Oracle VM VirtualBox". The menu bar includes "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". The terminal interface has a top bar with "Обзор" and "Терминал" tabs, and a timestamp "Вт, 17 сентября 13:14". The terminal content shows a user "guest2@ismakhorin:/home/guest" switching to root with "su -", entering a password, running "chmod +t /tmp", and then exiting back to the guest user. The word "выход" is also visible.

```
ismakhorin [Работает] - Oracle VM VirtualBox
Файл  Машина  Вид  Ввод  Устройства  Справка
Обзор  Терминал  Вт, 17 сентября 13:14
guest2@ismakhorin:/home/guest
guest@ismakhorin:/home/guest
[guest2@ismakhorin guest]$ su -
Пароль:
[root@ismakhorin ~]# chmod +t /tmp
[root@ismakhorin ~]# exit
выход
[guest2@ismakhorin guest]$
```

Рис. 3.28: Повышение прав до суперпользователя и возвращение атрибута t на директорию /tmp

4 Вывод

В ходе выполнения лабораторной работы были изучены механизмы изменения идентификаторов и применения SetUID- и Sticky-битов. Получены практические навыки работы в консоли с дополнительными атрибутами, а также была рассмотрена работа механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

5 Список литературы. Библиография

- [1] Дополнительные атрибуты: <https://tokmakov.msk.ru/blog/item/141>
- [2] Компилятор GSS: <http://parallel.imm.uran.ru/freesoft/make/instrum.html>