

Отчёт по лабораторной работе

Лабораторная №5

Панкратьев Александр Владимирович

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Исследование SetUID- и SetGID-битов	6
2.2	Исследование Sticky-бита	12
3	Вывод	14

List of Tables

List of Figures

2.1	Установка gcc	6
2.2	Снятие ограничений SELinux	6
2.3	Код программы simpleid.c	7
2.4	Компиляция и выполнение программы simpleid.c	7
2.5	Код программы simpleid2.c	8
2.6	Компиляция и выполнение программы simpleid2.c	8
2.7	Изменение атрибутов программы simpleid2	8
2.8	Вывод программы simpleid2 с атрибутом SetUID	9
2.9	Код программы readfile.c	9
2.10	Выполнение программы readfile	10
2.11	Смена атрибутов файла readfile.c	10
2.12	Проверка атрибутов файла readfile.c	11
2.13	Добавление SetUID-бита к программе readfile	11
2.14	Чтение файла readfile.c с помощью readfile	12
2.15	Проверка атрибута Sticky и создание файла в /tmp	12
2.16	Выполнение операций над file01.txt от имени guest2	13
2.17	Снятие атрибута Sticky с /tmp	13

1 Цель работы

Изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-биты. 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 Выполнение лабораторной работы

Для выполнения работы, установил компилятор gcc и отключил защиту SELinux (рис. 2.1, 2.2).

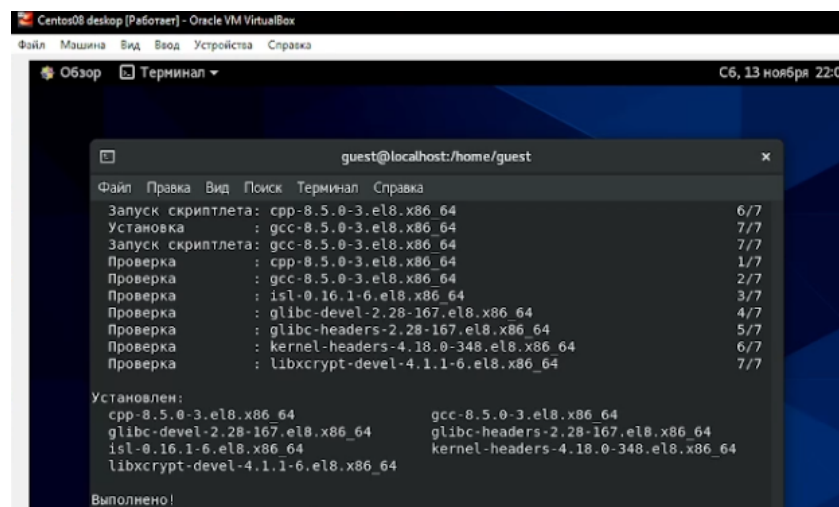


Figure 2.1: Установка gcc

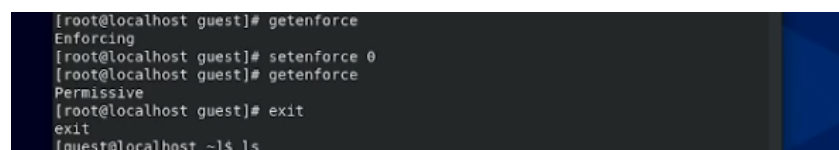


Figure 2.2: Снятие ограничений SELinux

2.1 Исследование SetUID- и SetGID-битов

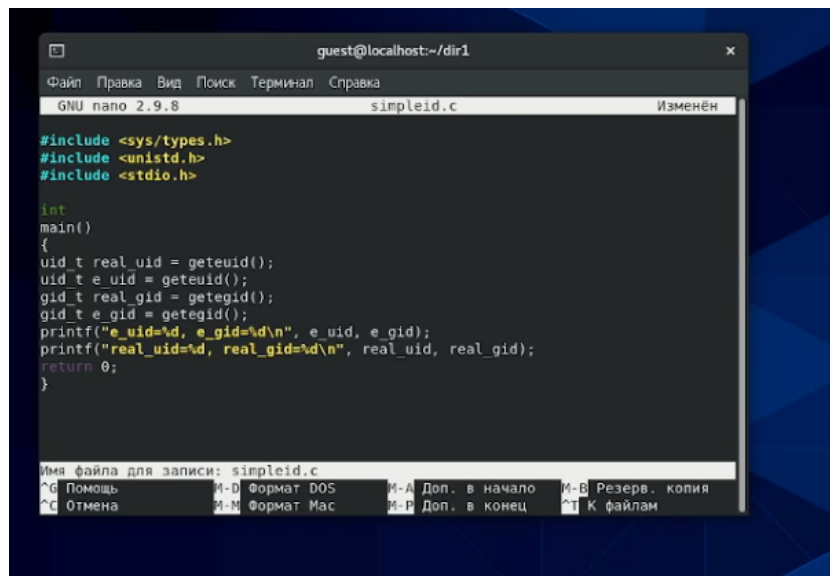
Вошёл в систему от пользователя guest и создал программу simpleid.c (рис. 2.3).

Figure 2.3: Код программы simpleid.c

Скомпилировал и выполнил программу. Полученный результат совпал с выводом команды id (рис. 2.4)

Figure 2.4: Компиляция и выполнение программы simpleid.c

Добавил в программу вывод действительных идентификаторов, назвал ее simpleid2.c (рис. 2.5).



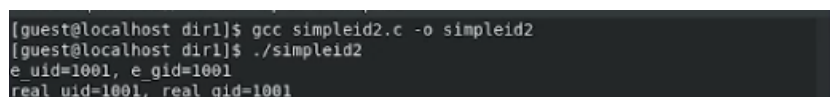
```
guest@localhost:~/dir1
Файл  Правка  Вид  Поиск  Терминал  Справка
GNU nano 2.9.8      simpleid.c  Изменён

#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t real_uid = geteuid();
    uid_t e_uid = geteuid();
    gid_t real_gid = getegid();
    gid_t e_gid = getegid();
    printf("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
    return 0;
}
```

Figure 2.5: Код программы simpleid2.c

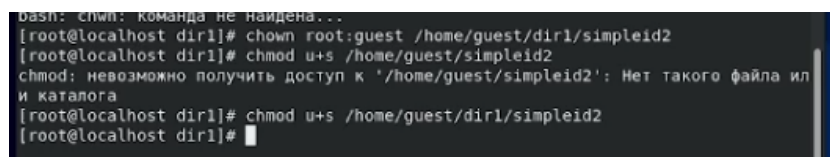
Скомпилировал и запустил программу simpleid2.c. Действительные идентификаторы совпали с эффективными (рис. 2.6)



```
[guest@localhost dir1]$ gcc simpleid2.c -o simpleid2
[guest@localhost dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

Figure 2.6: Компиляция и выполнение программы simpleid2.c

От имени суперпользователя изменила владельца программы simpleid2 на root и добавил атрибут SetUID. (рис. 2.7)



```
bash: chown: команда не найдена...
[root@localhost dir1]# chown root:guest /home/guest/dir1/simpleid2
[root@localhost dir1]# chown u+s /home/guest/simpleid2
chown: невозможно получить доступ к '/home/guest/simpleid2': Нет такого файла ил
и каталога
[root@localhost dir1]# chown u+s /home/guest/dir1/simpleid2
[root@localhost dir1]#
```

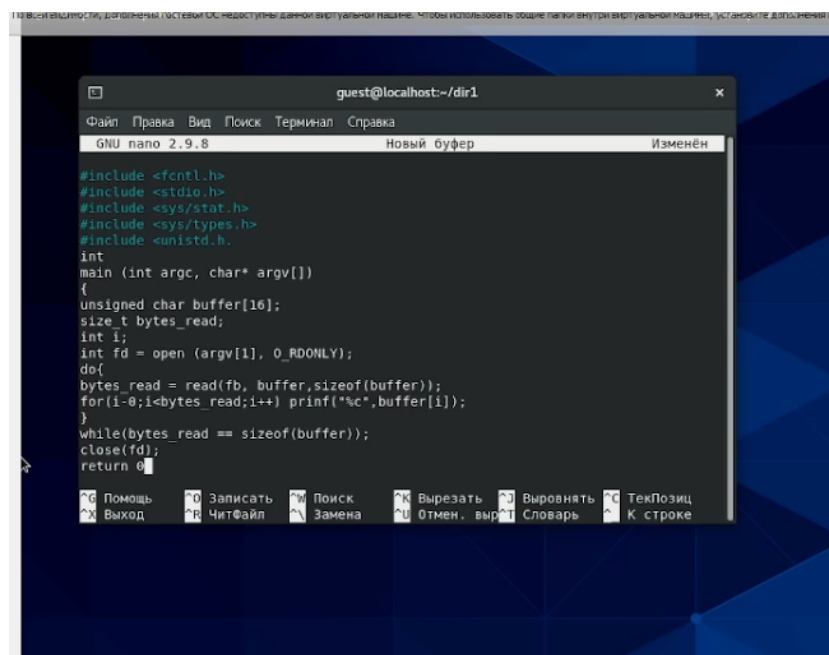
Figure 2.7: Изменение атрибутов программы simpleid2

Проверил правильность установки новых атрибутов и смены владельца файла simpleid2 и запустил simpleid2. Теперь вывод программы отличается от вывода команды id. Действительные идентификаторы остались прежними, а эффективный идентификатор пользователя теперь равен 0 - это идентификатор суперпользователя. Это значит, что пользователь guest использует права суперпользователя во время выполнения программы (рис. 2.8)


```
exit
[guest@localhost dir1]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 18144 ноя 13 22:11 simpleid2
[guest@localhost dir1]$ ./simpleid2
e uid=0, e_gid=1001
real uid=0, real_gid=1001
[guest@localhost dir1]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost dir1]$
```

Figure 2.8: Вывод программы simpleid2 с атрибутом SetUID

Создал программу readfile.c (рис. 2.9)



```
guest@localhost:~/dir1
GNU nano 2.9.8          Новый буфер          Изменен
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;
    int fd = open (argv[1], O_RDONLY);
    do{
        bytes_read = read(fd, buffer, sizeof(buffer));
        for(i=0;i<bytes_read;i++) printf("%c",buffer[i]);
    }
    while(bytes_read == sizeof(buffer));
    close(fd);
    return 0;
}
^G Помощь  ^O Записать  ^W Поиск  ^K Вырезать  ^J Выводить  ^C ТекПозиц
^X Выход    ^R ЧитФайл  ^M Замена  ^U Отмен. выр  ^_ Словарь  ^_ К строке
```

Figure 2.9: Код программы readfile.c

Откомпилировал и проверил корректность выполнения программы (рис. 2.10)

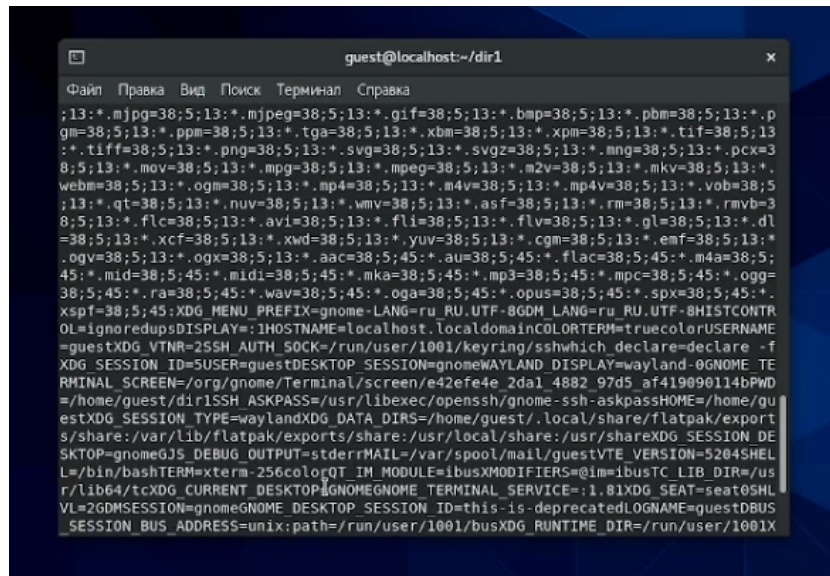


Figure 2.10: Выполнение программы readfile

Из-за неправильности системы, код не выполнялся

Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь мог прочитать его, а guest не мог (рис. 2.11)

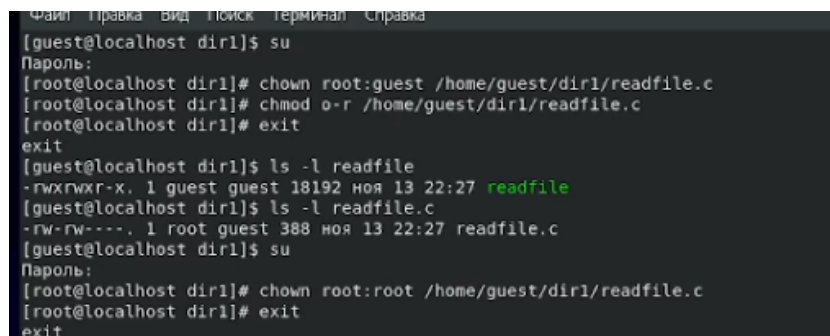


Figure 2.11: Смена атрибутов файла readfile.c

Проверил, что пользователь guest не может прочитать файл readfile.c (рис. 2.12)

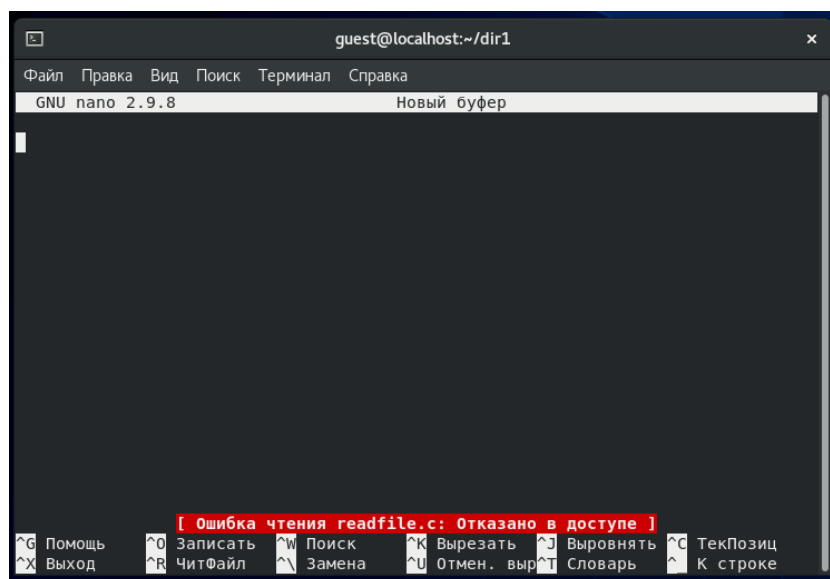


Figure 2.12: Проверка атрибутов файла readfile.c

Сменил у программы readfile владельца на root и установил SetUID-бит (рис. 2.13)

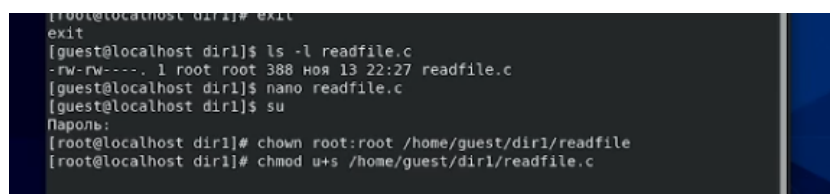
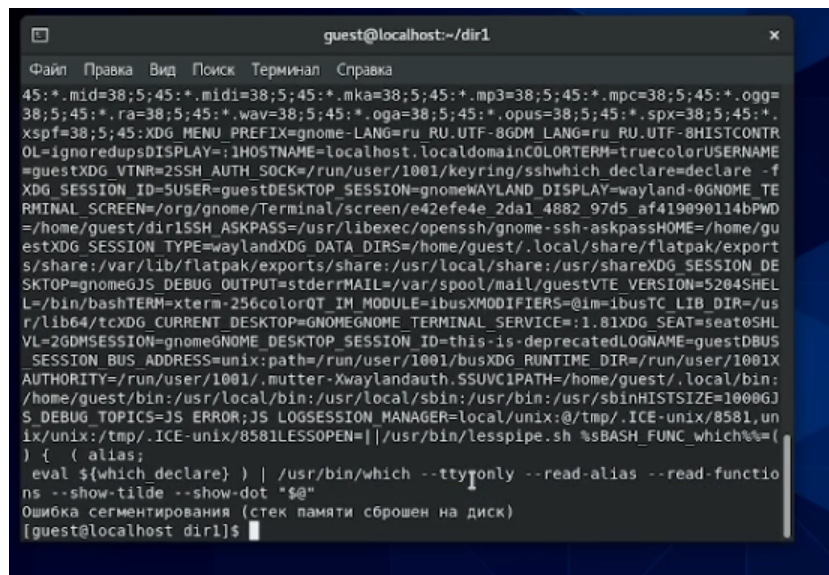


Figure 2.13: Добавление SetUID-бита к программе readfile

Теперь с помощью программы readfile можно от имени пользователя guest прочитать файл readfile.c. (рис. 2.14)

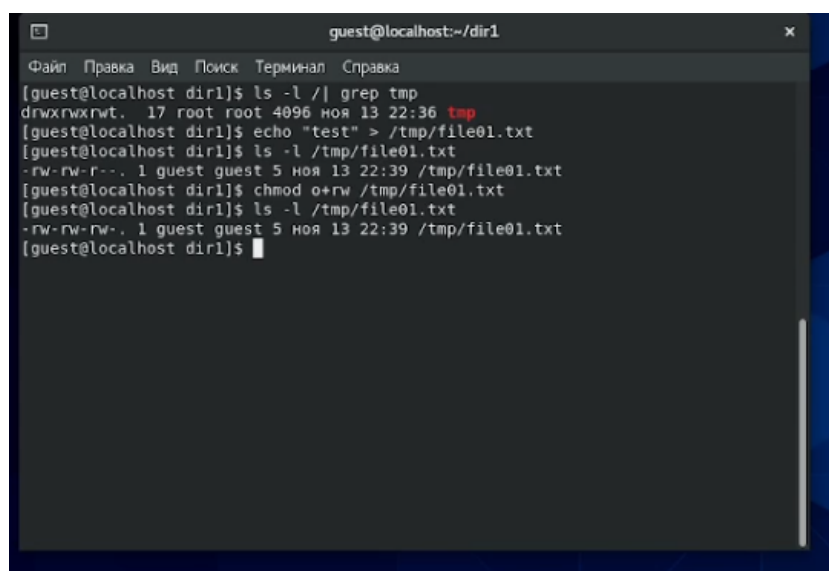


```
guest@localhost:~/dir1
Файл Правка Вид Поиск Терминал Справка
45:*.mid=38;5;45:*.midi=38;5;45:*.mka=38;5;45:*.mp3=38;5;45:*.mpc=38;5;45:*.ogg=
38;5;45:*.ra=38;5;45:*.wav=38;5;45:*.oga=38;5;45:*.opus=38;5;45:*.spx=38;5;45:*.
xspf=38;5;45:XDGMENU_PREFIX=gnome-LANG=ru RU.UTF-8GDM_LANG=ru RU.UTF-8HISTCONTR
OL=ignoreupsDISPLAY=:1HOSTNAME=localhost.localdomainCOLORTERM=truecolorUSERNAME
=guestXDGVTEVTNR=2SSH_AUTH_SOCK=/run/user/1001/keyring/sshwhich declare=declare -f
XDGMSESSION_ID=5USER=guestDESKTOP_SESSION=gnomeWAYLAND_DISPLAY=wayland-0GNOME TE
RMINAL_SCREEN=/org/gnome/terminal/screen/e42efe4e_2da1_4882_97d5_af419090114bPWD
=/home/guest/dir1SSH_ASKPASS=/usr/libexec/openssh/gnome-ssh-askpassHOME=/home/gu
estXDGMSESSION_TYPE=waylandXDGMDATA_DIRS=/home/guest/.local/share/flatpak/export
s/share:/var/lib/flatpak/exports/share:/usr/local/share:/usr/shareXDGMSESSION_DE
SKTOP=gnomeGJS_DEBUG_OUTPUT=stderrMAIL=/var/spool/mail/guestVTE_VERSION=5204SHEL
L=/bin/bashTERM=xterm-256colorQT_IM_MODULE=ibusXMODIFIERS=@im=ibusTC_LIB_DIR=/us
r/lib64/tcxdg_CURRENT_DESKTOP=GNOMEGNOME_TERMINAL_SERVICE=:1.81XDGMSEAT=seat0SHL
VL=2GDMSESSION=gnomeGNOME_DESKTOP_SESSION_ID=this-is-deprecatedLOGNAME=guestDBUS
_SESSION_BUS_ADDRESS=unix:path=/run/user/1001/busXDGM_RUNTIME_DIR=/run/user/1001X
AUTHORITY=/run/user/1001/.mutter-Xwaylandauth.SSUVCI_PATH=/home/guest/.local/bin:
/home/guest/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbinHISTSIZE=1000GJ
S_DEBUG_TOPICS=JS ERROR;JS LOGSESSION_MANAGER=local/unix:@/tmp/.ICE-unix/8581,un
ix/unix:/tmp/.ICE-unix/8581LESSOPEN=||/usr/bin/lesspipe.sh %sBASH_FUNC_which%=(
) { ( alias;
eval ${which declare} ) | /usr/bin/which --ttyonly --read-alias --read-functio
ns --show-tilde --show-dot "$@"
Ошибка сегментирования (стек памяти сброшен на диск)
[guest@localhost dir1]$
```

Figure 2.14: Чтение файла readfile.c с помощью readfile

2.2 Исследование Sticky-бита

Посмотрел, что на директории /tmp установлен атрибут Sticky. От имени пользователя guest создал файл file01.txt в директории /tmp со словом “test”. Посмотрел атрибуты у file01.txt и разрешил чтение и запись для категории пользователей “other” (рис. 2.17)



```
guest@localhost:~/dir1
Файл Правка Вид Поиск Терминал Справка
[guest@localhost dir1]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 ноя 13 22:36 tmp
[guest@localhost dir1]$ echo "test" > /tmp/file01.txt
[guest@localhost dir1]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 13 22:39 /tmp/file01.txt
[guest@localhost dir1]$ chmod o+rw /tmp/file01.txt
[guest@localhost dir1]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 13 22:39 /tmp/file01.txt
[guest@localhost dir1]$
```

Figure 2.15: Проверка атрибута Sticky и создание файла в /tmp

От пользователя guest попробовал выполнить различные действия - прочитать файл, дозаписать текст в файл, переписать текст в файле, удалить файл. Получилось сделать все (рис. 2.16)

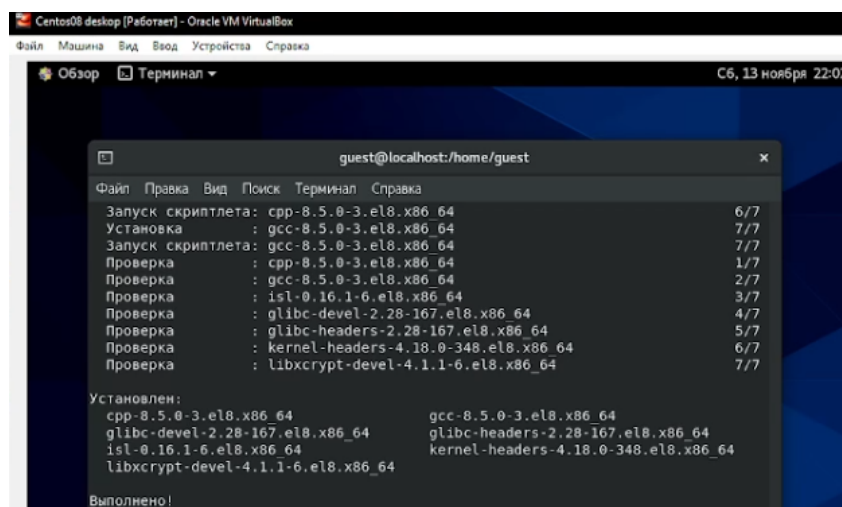


Figure 2.16: Выполнение операций над file01.txt от имени guest2

От имени суперпользователя снял Sticky-бит с директории /tmp (рис. 2.17)



Figure 2.17: Снятие атрибута Sticky с /tmp

3 Вывод

Я изучил механизмы изменения идентификаторов, применения SetUID- и Sticky-биты. Получил практические навыки работы в консоли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.