

Отчёт по лабораторной работе №5

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

Абузярова Лейла Джамилевна НБИбд-01-19

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Подготовка	5
2.2	Изучение механики SetUID	7
2.3	Исследование Sticky-бита	14
3	Выводы	17
	Список литературы	18

List of Figures

2.1	Установка компилятора gcc	5
2.2	Подготовка к работе	6
2.3	Проверка установки компилятора	6
2.4	Создание и редактирование файла simpleid.c	7
2.5	Написание программы	8
2.6	Результат программы simpleid	9
2.7	Создание и редактирование файла simpleid2.c	9
2.8	Написание программы	10
2.9	Результат программы simpleid2 от guest	10
2.10	Результат программы simpleid2 от root	11
2.11	Создание и редактирование файла readfile.c	12
2.12	Программа readfile	12
2.13	Смена прав на файле readfile	13
2.14	Проверка файла на чтение	13
2.15	Повторная проверка файла на чтение	14
2.16	Проверка прав на файле file01.txt	15
2.17	Дозапись и перезапись слов в файл file01.txt	15
2.18	Снятие атрибута t	16

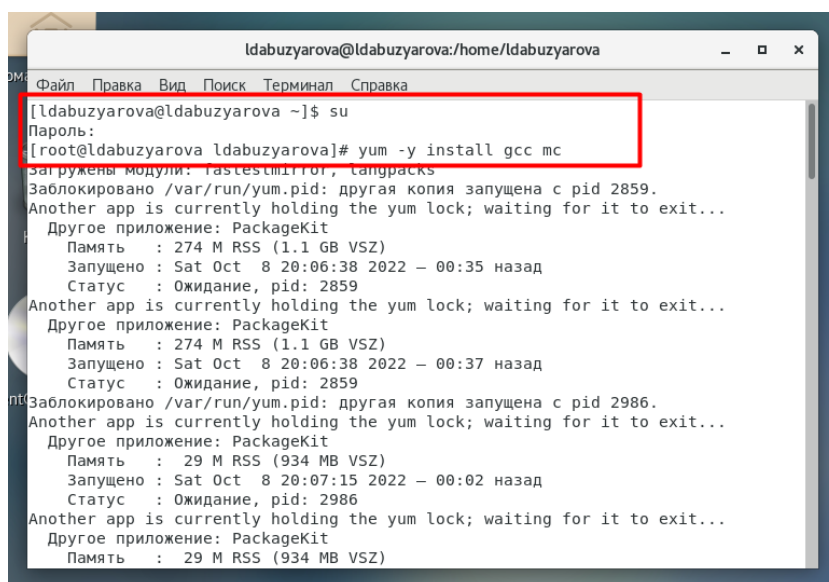
1 Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

2 Выполнение лабораторной работы

2.1 Подготовка

Для выполнения части заданий требуются средства разработки приложений. Поэтому установим компилятор gcc командой `yum install gcc`.



```
ldabuzyarova@ldabuzyarova:/home/ldabuzyarova
[ldabuzyarova@ldabuzyarova ~]$ su
Пароль:
[root@ldabuzyarova ldabuzyarova]# yum -y install gcc mc
Загружены модули: fastestmirror, langpacks
Заблокировано /var/run/yum.pid: другая копия запущена с pid 2859.
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 274 M RSS (1.1 GB VSZ)
Запущено : Sat Oct 8 20:06:38 2022 - 00:35 назад
Статус : Ожидание, pid: 2859
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 274 M RSS (1.1 GB VSZ)
Запущено : Sat Oct 8 20:06:38 2022 - 00:37 назад
Статус : Ожидание, pid: 2859
Заблокировано /var/run/yum.pid: другая копия запущена с pid 2986.
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 29 M RSS (934 MB VSZ)
Запущено : Sat Oct 8 20:07:15 2022 - 00:02 назад
Статус : Ожидание, pid: 2986
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 29 M RSS (934 MB VSZ)
```

Figure 2.1: Установка компилятора gcc

Чтобы система защиты SELinux не мешала выполнению заданий работы, отключили систему запретов до очередной перезагрузки системы командой `setenforce 0`. Команда `getenforce` вывела `Permissive`:

```
ldabuzyarova@ldabuzyarova:/home/ldabuzyarova
Файл Правка Вид Поиск Терминал Справка
Установка : kernel-headers-3.10.0-1160.76.1.el7.x86_64 2/5
Установка : glibc-headers-2.17-326.el7_9.x86_64 3/5
Установка : glibc-devel-2.17-326.el7_9.x86_64 4/5
Установка : gcc-4.8.5-44.el7.x86_64 5/5
Проверка : glibc-headers-2.17-326.el7_9.x86_64 1/5
Проверка : gcc-4.8.5-44.el7.x86_64 2/5
Проверка : kernel-headers-3.10.0-1160.76.1.el7.x86_64 3/5
Проверка : glibc-devel-2.17-326.el7_9.x86_64 4/5
Проверка : cpp-4.8.5-44.el7.x86_64 5/5

Установлено:
gcc.x86_64 0:4.8.5-44.el7

Установлены зависимости:
cpp.x86_64 0:4.8.5-44.el7
glibc-devel.x86_64 0:2.17-326.el7_9
glibc-headers.x86_64 0:2.17-326.el7_9
kernel-headers.x86_64 0:3.10.0-1160.76.1.el7

Выполнено!
[root@ldabuzyarova ldabuzyarova]# setenforce 0
[root@ldabuzyarova ldabuzyarova]# getenforce
Permissive
[root@ldabuzyarova ldabuzyarova]#
```

Figure 2.2: Подготовка к работе

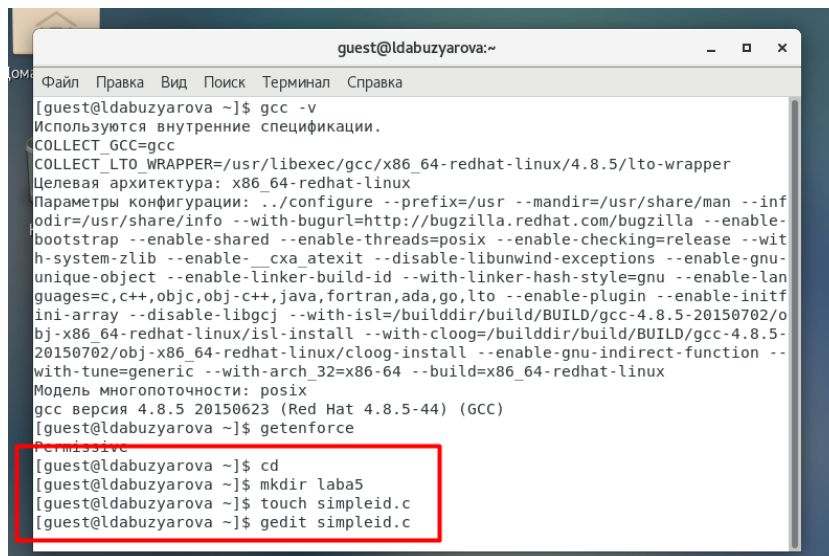
Теперь проверим корректность установки компилятора, как показано на скриншоте номер 3.

```
guest@ldabuzyarova:~
Файл Правка Вид Поиск Терминал Справка
[guest@ldabuzyarova ~]$ gcc -v
используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-cxx --enable-languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версии 4.8.5-20150623 (Red Hat 4.8.5-44) (GCC)
[guest@ldabuzyarova ~]$ getenforce
Permissive
[guest@ldabuzyarova ~]$
```

Figure 2.3: Проверка установки компилятора

2.2 Изучение механики SetUID

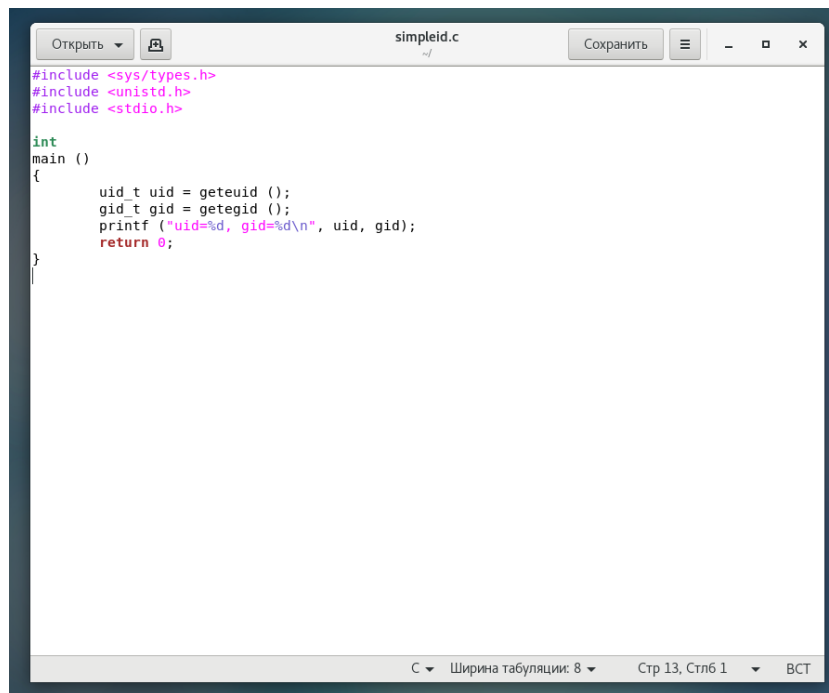
Вошли в систему от имени пользователя guest. Создали папку laba5, создали файл simpleid.c. Начали редактирование данного файла.



```
guest@dabuzyarova:~  
[guest@dabuzyarova ~]$ gcc -v  
Используются внутренние спецификации.  
COLLECT_GCC=gcc  
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper  
Целевая архитектура: x86_64-redhat-linux  
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-bootstrap --enable-shared --enable-threads=posix --enable-checking=release --with-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initfini-array --disable-libgcj --with-isl=/build/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/isl-install --with-cloog=/build/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux  
Модель многопоточности: posix  
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[guest@dabuzyarova ~]$ getenforce  
Permissive  
[guest@dabuzyarova ~]$ cd  
[guest@dabuzyarova ~]$ mkdir laba5  
[guest@dabuzyarova ~]$ touch simpleid.c  
[guest@dabuzyarova ~]$ gedit simpleid.c
```

Figure 2.4: Создание и редактирование файла simpleid.c

Написали следующую программу в файле simpleid.c.

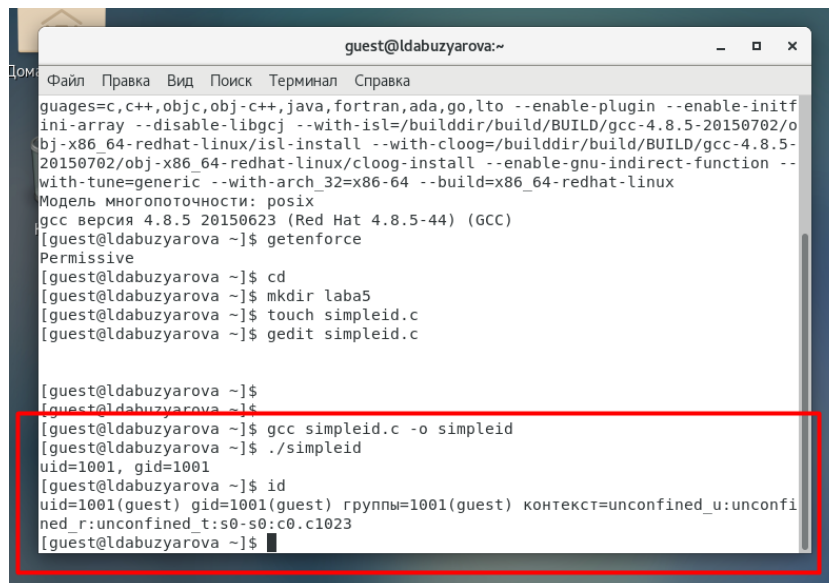


```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main ()
{
    uid_t uid = getuid ();
    gid_t gid = getgid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
```

Figure 2.5: Написание программы

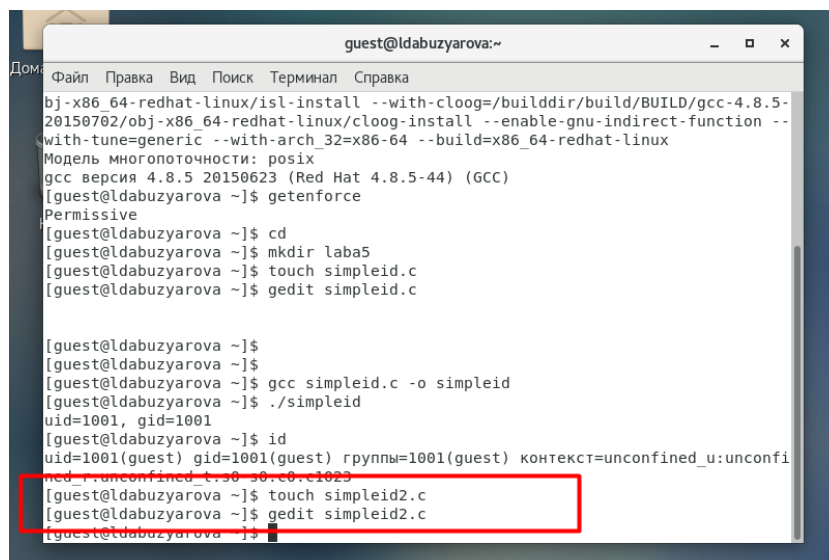
Скомпилировали программу и убедились, что файл программы создан: `gcc simpleid.c -o simpleid`. Выполнили программу `simpleid` командой `./simpleid`. Выполнили системную программу `id` с помощью команды `id`. `uid` и `gid` совпадает в обеих программах



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
guages=c, c++, obj-c, obj-c++, java, fortran, ada, go, lto --enable-plugin --enable-initf  
ini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o  
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-  
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --  
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux  
Модель многопоточности: posix  
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[guest@ldabuzyarova ~]$ getenforce  
Permissive  
[guest@ldabuzyarova ~]$ cd  
[guest@ldabuzyarova ~]$ mkdir laba5  
[guest@ldabuzyarova ~]$ touch simpleid.c  
[guest@ldabuzyarova ~]$ gedit simpleid.c  
  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ gcc simpleid.c -o simpleid  
[guest@ldabuzyarova ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@ldabuzyarova ~]$ id  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@ldabuzyarova ~]$
```

Figure 2.6: Результат программы simpleid

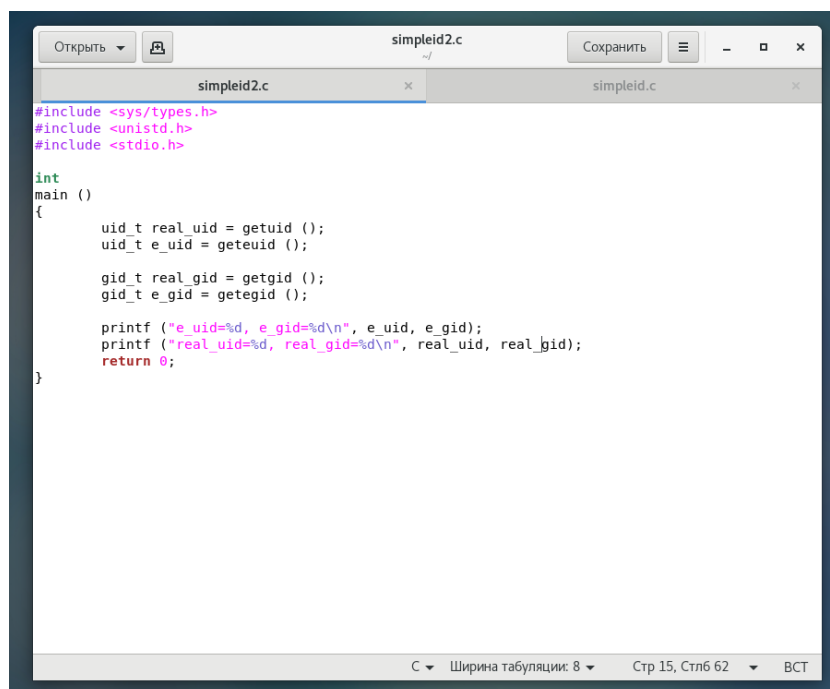
Усложнили программу, добавив вывод действительных идентификаторов. Про-
делываем те же операции, что и с предыдущим файлом.



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-  
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --  
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux  
Модель многопоточности: posix  
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[guest@ldabuzyarova ~]$ getenforce  
Permissive  
[guest@ldabuzyarova ~]$ cd  
[guest@ldabuzyarova ~]$ mkdir laba5  
[guest@ldabuzyarova ~]$ touch simpleid.c  
[guest@ldabuzyarova ~]$ gedit simpleid.c  
  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ gcc simpleid.c -o simpleid  
[guest@ldabuzyarova ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@ldabuzyarova ~]$ id  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@ldabuzyarova ~]$ touch simpleid2.c  
[guest@ldabuzyarova ~]$ gedit simpleid2.c  
[guest@ldabuzyarova ~]$
```

Figure 2.7: Создание и редактирование файла simpleid2.c

Написали следующую программу в файле simpleid2.c.



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

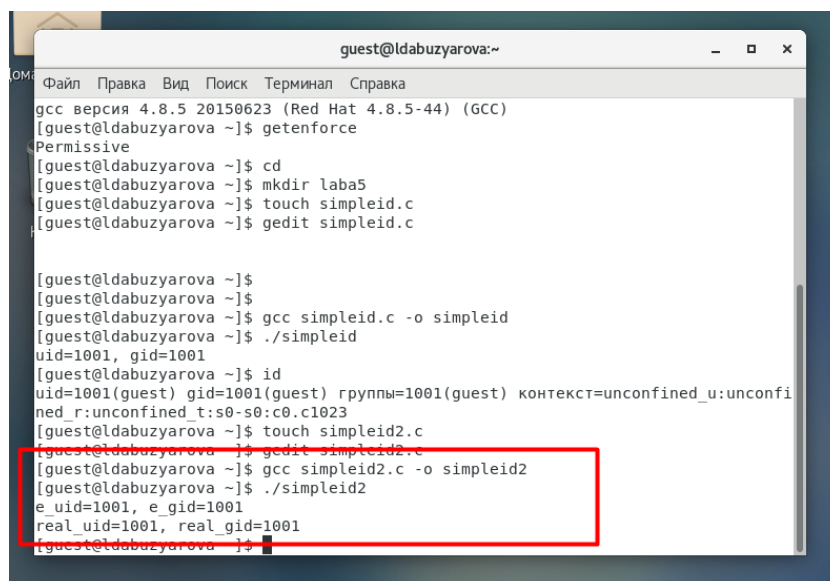
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();

    gid_t real_gid = getgid ();
    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 2.8: Написание программы

Скомпилировали и запустили simpleid2.c.



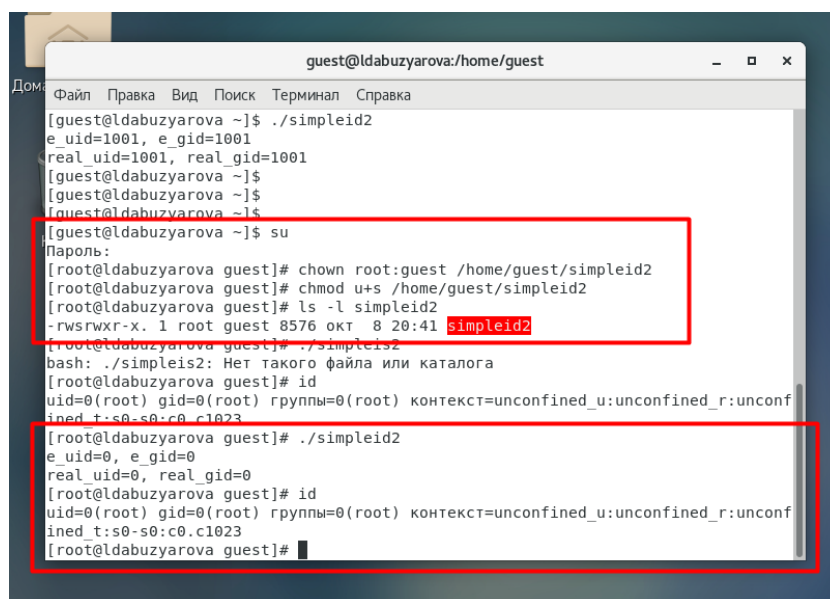
```
guest@dabuzyarova:~
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[guest@dabuzyarova ~]$ getenforce
Permissive
[guest@dabuzyarova ~]$ cd
[guest@dabuzyarova ~]$ mkdir laba5
[guest@dabuzyarova ~]$ touch simpleid.c
[guest@dabuzyarova ~]$ gedit simpleid.c

[guest@dabuzyarova ~]$
[guest@dabuzyarova ~]$
[guest@dabuzyarova ~]$ gcc simpleid.c -o simpleid
[guest@dabuzyarova ~]$ ./simpleid
uid=1001, gid=1001
[guest@dabuzyarova ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@dabuzyarova ~]$ touch simpleid2.c
[guest@dabuzyarova ~]$ gedit simpleid2.c
[guest@dabuzyarova ~]$ gcc simpleid2.c -o simpleid2
[guest@dabuzyarova ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@dabuzyarova ~]$
```

Figure 2.9: Результат программы simpleid2 от guest

На скриншоте 10 видно следующее: теперь от имени суперпользователя выполнили команды, которые меняют владельца файла. Использовали su для повыше-

ния прав до суперпользователя. Выполнили проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Также запустили simpleid2 и id.

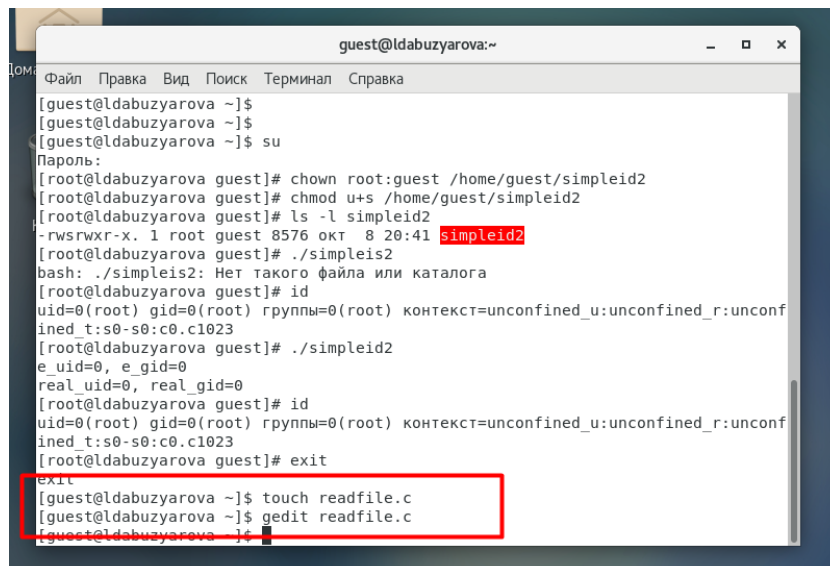


```
guest@ldabuzyarova:/home/guest
[guest@ldabuzyarova ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real uid=1001, real_gid=1001
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$ su
Пароль:
[root@ldabuzyarova guest]# chown root:guest /home/guest/simpleid2
[root@ldabuzyarova guest]# chmod u+s /home/guest/simpleid2
[root@ldabuzyarova guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 окт  8 20:41 simpleid2
[root@ldabuzyarova guest]# ./simpleid2
bash: ./simpleid2: Нет такого файла или каталога
[root@ldabuzyarova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ldabuzyarova guest]# ./simpleid2
e_uid=0, e_gid=0
real uid=0, real_gid=0
[root@ldabuzyarova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ldabuzyarova guest]#
```

Figure 2.10: Результат программы simpleid2 от root

Приходим к выводу, что результат выполнения программ при заходе от пользователя guest и через root отличается.

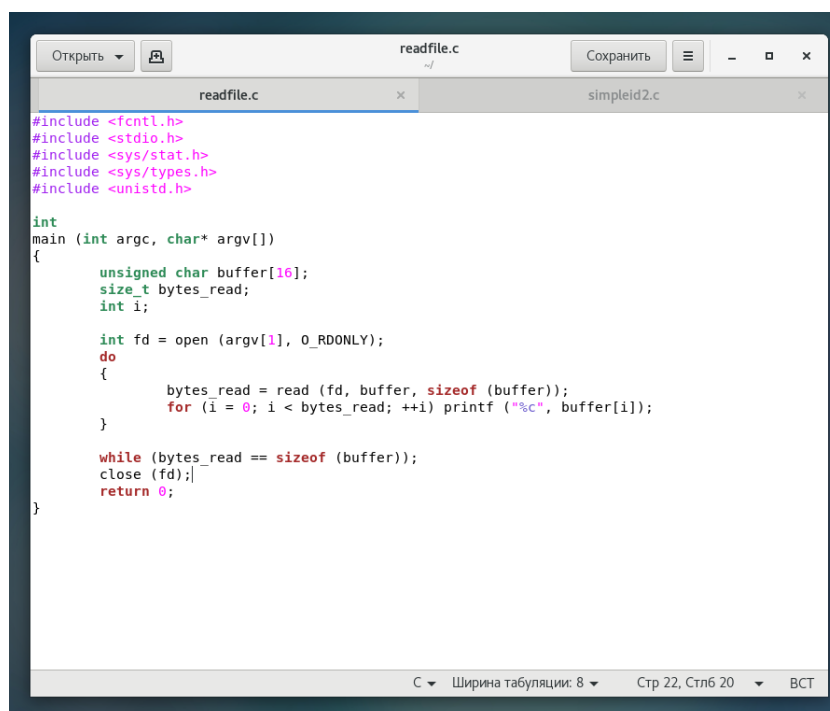
Теперь создадим файл readfile.c, применяя те же команды, что и в предыдущий раз.

A terminal window titled 'guest@ldabuzyarova:~' showing a series of commands. The user switches to root, sets permissions on /home/guest/simpleid2, and then creates and edits a file named readfile.c. The file's permissions are shown as -rwsrwxr-x. The user then exits root and uses 'touch' and 'gedit' to create and edit readfile.c. The last two commands are highlighted with a red box.

```
guest@ldabuzyarova:~$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# chown root:guest /home/guest/simpleid2  
[root@ldabuzyarova guest]# chmod u+s /home/guest/simpleid2  
[root@ldabuzyarova guest]# ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8576 окт  8 20:41 simpleid2  
[root@ldabuzyarova guest]# ./simpleid2  
bash: ./simpleid2: Нет такого файла или каталога  
[root@ldabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@ldabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ touch readfile.c  
[guest@ldabuzyarova ~]$ gedit readfile.c  
[guest@ldabuzyarova ~]$
```

Figure 2.11: Создание и редактирование файла readfile.c

Написали программу readfile.c

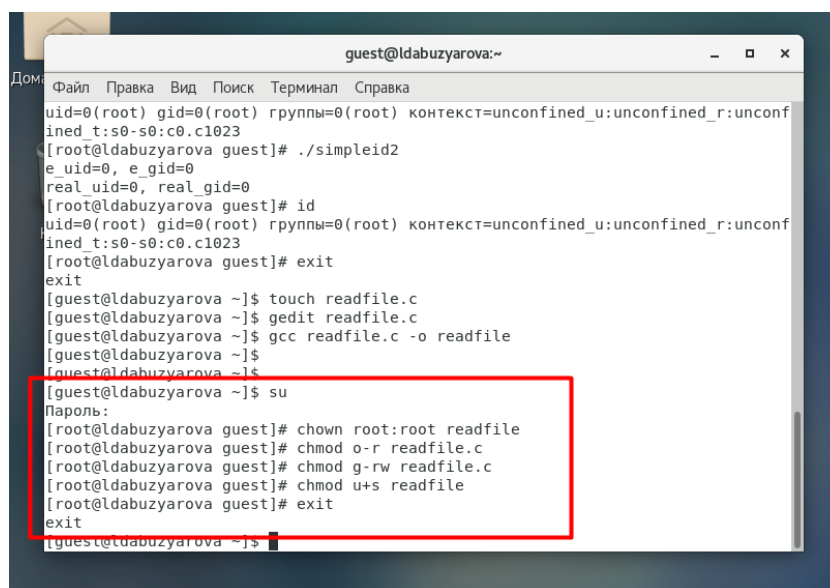
A screenshot of a code editor showing the source code of the readfile.c program. The code includes headers for `<fcntl.h>`, `<stdio.h>`, `<sys/stat.h>`, `<sys/types.h>`, and `<unistd.h>`. It defines a `main` function that takes `argc` and `argv` as arguments. Inside the function, it declares a `buffer` of size 16, a `bytes_read` variable, and an `i` index. It opens the file specified in `argv[1]` in read-only mode, reads it into the buffer, and prints each byte. The program returns 0 after closing the file.

```
readfile.c  
#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;  
}
```

Figure 2.12: Программа readfile

Откомпилировали программу. Затем сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, а

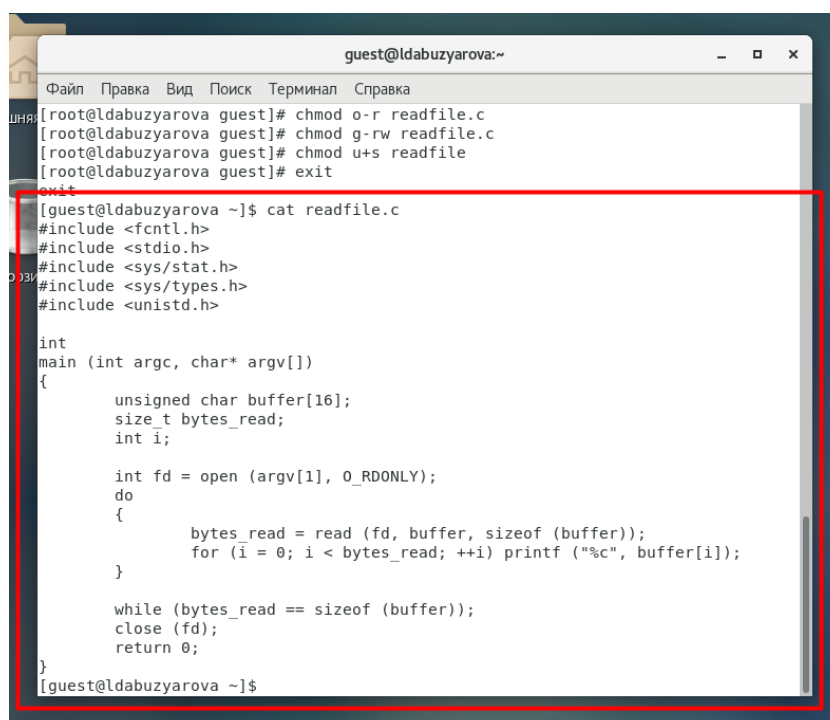
guest не мог.



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@ldabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ touch readfile.c  
[guest@ldabuzyarova ~]$ gedit readfile.c  
[guest@ldabuzyarova ~]$ gcc readfile.c -o readfile  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# chown root:root readfile  
[root@ldabuzyarova guest]# chmod o-r readfile.c  
[root@ldabuzyarova guest]# chmod g-rw readfile.c  
[root@ldabuzyarova guest]# chmod u+s readfile  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$
```

Figure 2.13: Смена прав на файле readfile

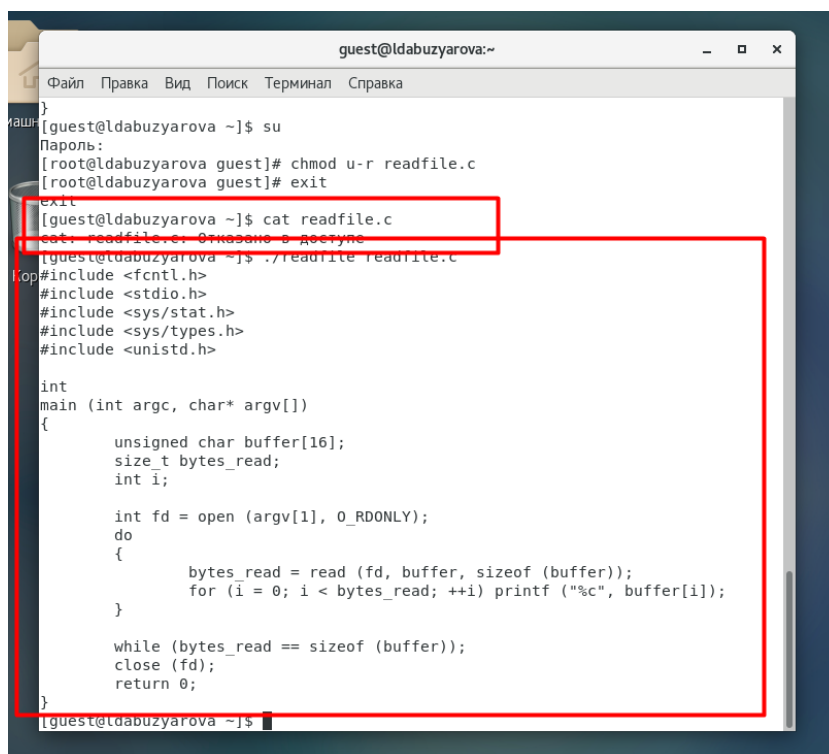
Проверили, что пользователь guest не может прочитать файл readfile.c. Но увидели, что файл читается.



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
[root@ldabuzyarova guest]# chmod o-r readfile.c  
[root@ldabuzyarova guest]# chmod g-rw readfile.c  
[root@ldabuzyarova guest]# chmod u+s readfile  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ cat readfile.c  
#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;  
}  
[guest@ldabuzyarova ~]$
```

Figure 2.14: Проверка файла на чтение

Ищем ошибку. Оказалось, что была допущена ошибка в установке прав. Переделали команду с установкой прав и повторили проверку, на этот раз в доступе отказано, как и должно быть. А команда `./readfile` сработала потому что она выполняет чтение от имени суперпользователя, а не `guest`.

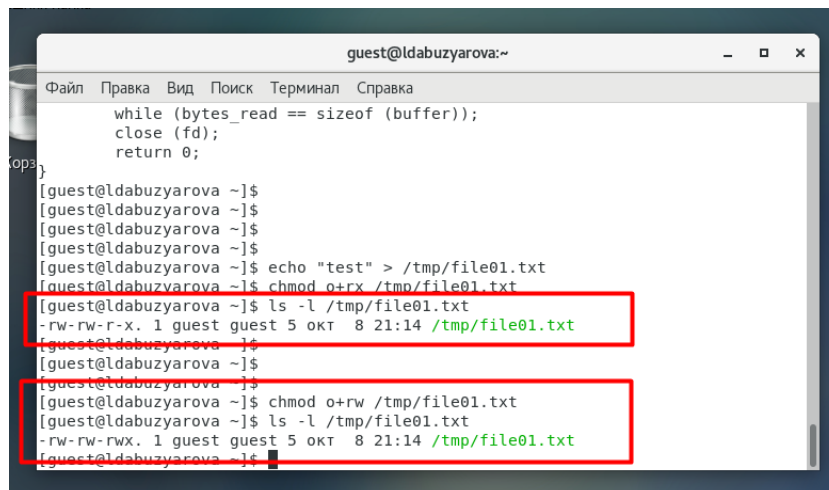


```
guest@ldabuzyarova:~  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# chmod u-r readfile.c  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ cat readfile.c  
cat: readfile.c: Отказано в доступе  
[guest@ldabuzyarova ~]$ ./readfile readfile.c  
#include <fcntl.h>  
#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;  
}
```

Figure 2.15: Повторная проверка файла на чтение

2.3 Исследование Sticky-бита

От имени пользователя `guest` создали файл `file01.txt` в директории `/tmp` со словом `test`. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». Первоначально все группы имели право на чтение, а запись могли осуществлять все, кроме «остальных пользователей».

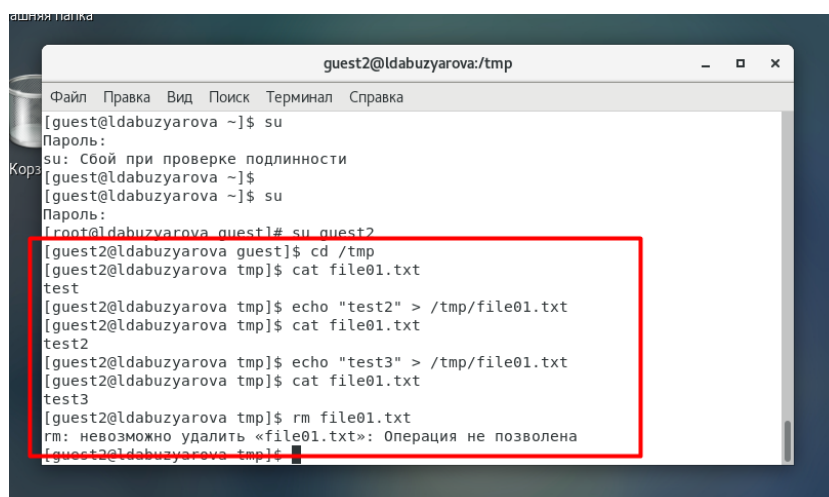
A terminal window titled 'guest@ldabuzyarova:~' showing a series of commands and their outputs. The commands are: 'while (bytes_read == sizeof (buffer)); close (fd); return 0;}', '[guest@ldabuzyarova ~]\$', '[guest@ldabuzyarova ~]\$', '[guest@ldabuzyarova ~]\$', '[guest@ldabuzyarova ~]\$', '[guest@ldabuzyarova ~]\$ echo "test" > /tmp/file01.txt', '[guest@ldabuzyarova ~]\$ chmod o+rx /tmp/file01.txt', '[guest@ldabuzyarova ~]\$ ls -l /tmp/file01.txt', and '[guest@ldabuzyarova ~]\$'. The output of the 'ls' command is highlighted with a red box: '-rw-rw-r-x. 1 guest guest 5 окт 8 21:14 /tmp/file01.txt'.

```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
while (bytes_read == sizeof (buffer));  
close (fd);  
return 0;  
}  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ echo "test" > /tmp/file01.txt  
[guest@ldabuzyarova ~]$ chmod o+rx /tmp/file01.txt  
[guest@ldabuzyarova ~]$ ls -l /tmp/file01.txt  
-rw-rw-r-x. 1 guest guest 5 окт 8 21:14 /tmp/file01.txt  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ chmod o+rw /tmp/file01.txt  
[guest@ldabuzyarova ~]$ ls -l /tmp/file01.txt  
-rw-rw-rwx. 1 guest guest 5 окт 8 21:14 /tmp/file01.txt  
[guest@ldabuzyarova ~]$
```

Figure 2.16: Проверка прав на файле file01.txt

От пользователя (не являющегося владельцем) попробовали прочитать файл /file01.txt. Мы видим, что в файле записано одно слово. далее к нему мы должны были поочередно дозаписать слова test2 и test3, но я забыла добавить по одной галочке и вместо дозаписи перезаписала слова.

От пользователя попробовали удалить файл /tmp/file01.txt командой rm /tmp/file01.txt, однако получила отказ.

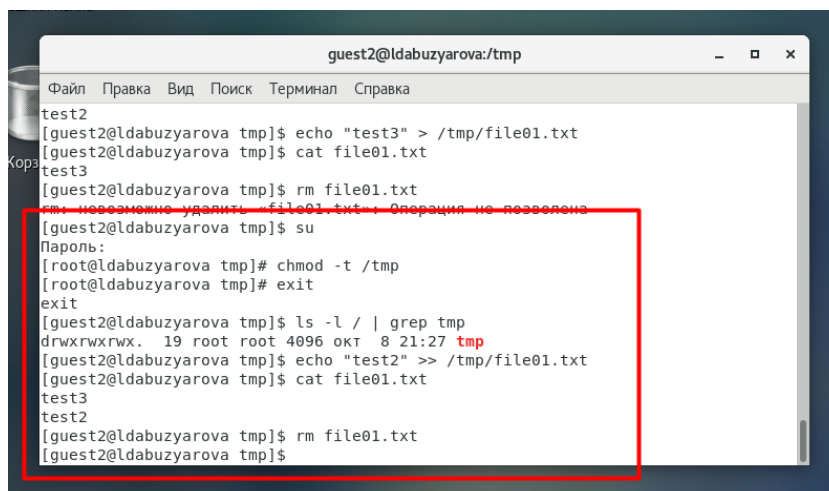
A terminal window titled 'guest2@ldabuzyarova:/tmp' showing a series of commands and their outputs. The commands are: '[guest@ldabuzyarova ~]\$ su', 'Пароль:', 'su: Сбой при проверке подлинности', '[guest@ldabuzyarova ~]\$', '[guest@ldabuzyarova ~]\$ su', 'Пароль:', '[root@ldabuzyarova guest]# su guest2', '[guest2@ldabuzyarova guest]\$ cd /tmp', '[guest2@ldabuzyarova tmp]\$ cat file01.txt', '[guest2@ldabuzyarova tmp]\$ echo "test2" > /tmp/file01.txt', '[guest2@ldabuzyarova tmp]\$ cat file01.txt', '[guest2@ldabuzyarova tmp]\$ echo "test3" > /tmp/file01.txt', '[guest2@ldabuzyarova tmp]\$ cat file01.txt', '[guest2@ldabuzyarova tmp]\$ rm file01.txt', and '[guest2@ldabuzyarova tmp]\$'. The output of the 'cat' and 'rm' commands is highlighted with a red box: 'test', 'test2', 'test3', and 'rm: невозможно удалить «file01.txt»: Операция не позволена'.

```
guest2@ldabuzyarova:/tmp  
Файл Правка Вид Поиск Терминал Справка  
[guest@ldabuzyarova ~]$ su  
Пароль:  
su: Сбой при проверке подлинности  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# su guest2  
[guest2@ldabuzyarova guest]$ cd /tmp  
[guest2@ldabuzyarova tmp]$ cat file01.txt  
test  
[guest2@ldabuzyarova tmp]$ echo "test2" > /tmp/file01.txt  
[guest2@ldabuzyarova tmp]$ cat file01.txt  
test2  
[guest2@ldabuzyarova tmp]$ echo "test3" > /tmp/file01.txt  
[guest2@ldabuzyarova tmp]$ cat file01.txt  
test3  
[guest2@ldabuzyarova tmp]$ rm file01.txt  
rm: невозможно удалить «file01.txt»: Операция не позволена  
[guest2@ldabuzyarova tmp]$
```

Figure 2.17: Дозапись и перезапись слов в файл file01.txt

От суперпользователя командой выполнили команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Покинули режим суперпользователя командой

exit. От пользователя проверили, что атрибута t у директории /tmp нет. Повторили предыдущие шаги. Получилось удалить файл.



```
guest2@dabuzyarova:/tmp
Файл Правка Вид Поиск Терминал Справка
test2
[guest2@dabuzyarova tmp]$ echo "test3" > /tmp/file01.txt
[guest2@dabuzyarova tmp]$ cat file01.txt
test3
[guest2@dabuzyarova tmp]$ rm file01.txt
rm: невозможно удалить 'file01.txt': Операция не разрешена
[guest2@dabuzyarova tmp]$ su
Пароль:
[root@dabuzyarova tmp]# chmod -t /tmp
[root@dabuzyarova tmp]# exit
exit
[guest2@dabuzyarova tmp]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 окт 8 21:27 tmp
[guest2@dabuzyarova tmp]$ echo "test2" >> /tmp/file01.txt
[guest2@dabuzyarova tmp]$ cat file01.txt
test3
test2
[guest2@dabuzyarova tmp]$ rm file01.txt
[guest2@dabuzyarova tmp]$
```

Figure 2.18: Снятие атрибута t

3 Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.

Список литературы

1. КОМАНДА CHATTR В LINUX
2. chattr