

Лабораторная работа №5.

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

Силкина Мария Александровна

Содержание

1	Цель работы	6
2	Задачи	7
3	Выполнение лабораторной работы	8
3.1	Подготовка лабораторного стенда	8
3.2	Создание программ	9
3.3	Исследование Sticky-бита	13
4	Выводы	16
5	Библиография	17

List of Tables

List of Figures

3.1	Установка компилятора	8
3.2	Отключение системы запретов	9
3.3	Создание файла и его редактирование	9
3.4	Код программы	10
3.5	Компиляция	10
3.6	Сравнение вывода программ	10
3.7	Создание и редактирование файла	10
3.8	Код программы	11
3.9	Компиляция и вывод	11
3.10	Выполнение команд	11
3.11	Сравнение вывода программ	12
3.12	Создание файла	12
3.13	Код программы	12
3.14	Смен владельца	13
3.15	Изменение прав	13
3.16	Проверка	13
3.17	Установка SetUID-бита	13
3.18	Проверка	13
3.19	Проверка наличия атрибута, создание файла, установка прав на него	14
3.20	Выполнение действий	14
3.21	Удаление атрибута	14
3.22	Выполнение действий без атрибута Sticky	15

3.23 Возвращение атрибута t	15
---------------------------------------	----

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.

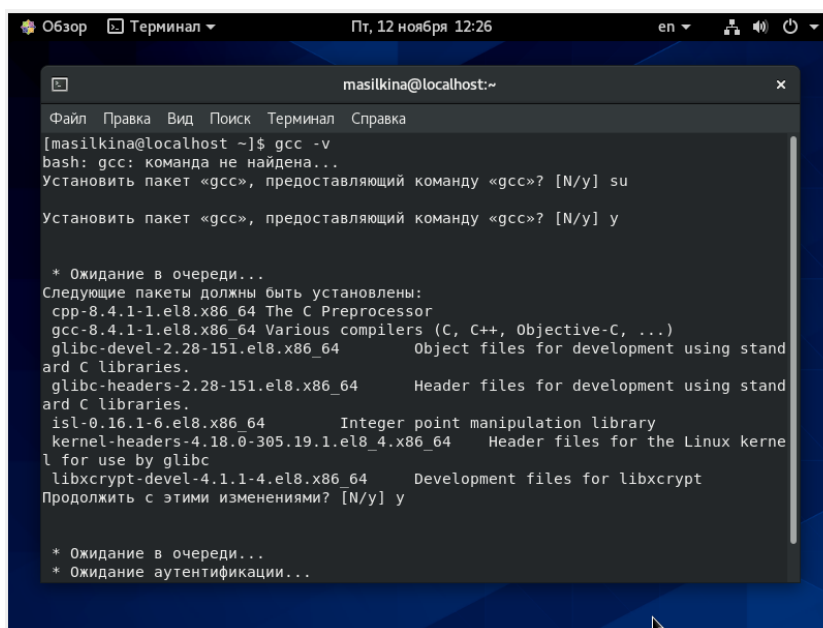
2 Задачи

1. Выполнить лабораторную работу согласно заданному порядку.
2. Ознакомится с применением SetUID- и Sticky-битов, изучение их влияния.

3 Выполнение лабораторной работы

3.1 Подготовка лабораторного стенда

Первый шаг заключался в установке компилятора gcc для дальнейшего выполнения лабораторной работы, а также отключении системы запретов (рис 1. @fig:001) (рис 2. @fig:002).

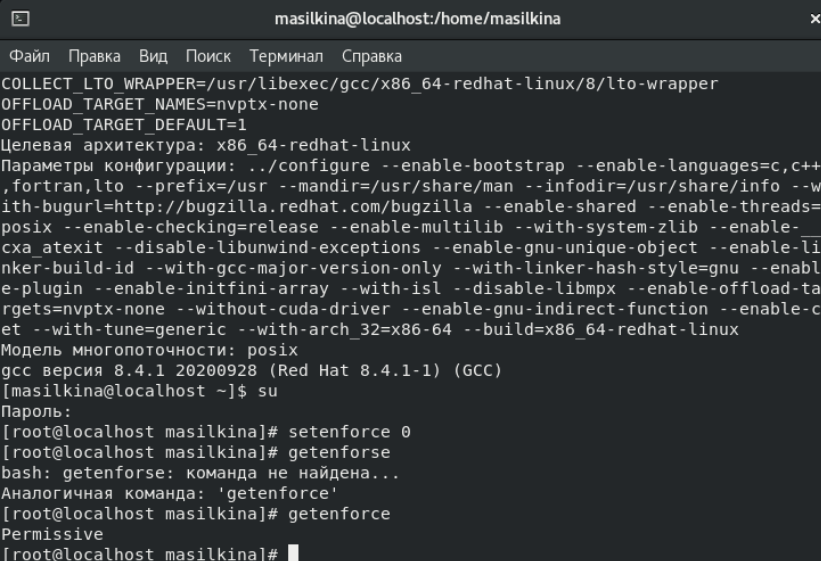


```
Обзор Терминал Пт, 12 ноября 12:26 en [system icons]
masilkina@localhost:~$ gcc -v
bash: gcc: команда не найдена...
Установить пакет «gcc», предоставляющий команду «gcc»? [N/y] su
Установить пакет «gcc», предоставляющий команду «gcc»? [N/y] y

* Ожидание в очереди...
Следующие пакеты должны быть установлены:
cpr-8.4.1-1.el8.x86_64 The C Preprocessor
gcc-8.4.1-1.el8.x86_64 Various compilers (C, C++, Objective-C, ...)
glibc-devel-2.28-151.el8.x86_64 Object files for development using stand
ard C libraries.
glibc-headers-2.28-151.el8.x86_64 Header files for development using stand
ard C libraries.
isl-0.16.1-6.el8.x86_64 Integer point manipulation library
kernel-headers-4.18.0-305.19.1.el8_4.x86_64 Header files for the Linux kerne
l for use by glibc
libxcrypt-devel-4.1.1-4.el8.x86_64 Development files for libxcrypt
Продолжить с этими изменениями? [N/y] y

* Ожидание в очереди...
* Ожидание аутентификации...
```

Figure 3.1: Установка компилятора

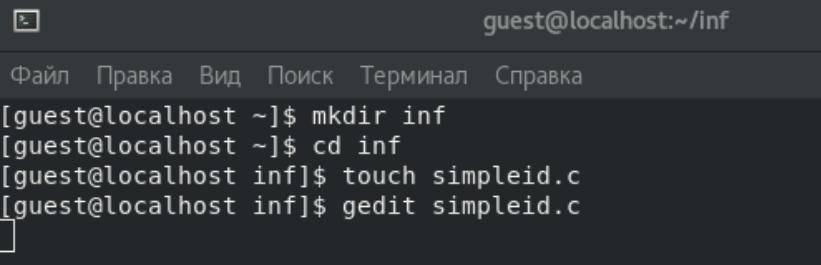


```
masilkina@localhost:/home/masilkina
Файл Правка Вид Поиск Терминал Справка
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/8/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-shared --enable-threads=posix --enable-checking=release --enable-multilib --with-system-zlib --enable-cxx-atomic --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --with-isl --disable-libmpx --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --enable-cet --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 8.4.1 20200928 (Red Hat 8.4.1-1) (GCC)
[masilkina@localhost ~]$ su
Пароль:
[root@localhost masilkina]# setenforce 0
[root@localhost masilkina]# getenforce
bash: getenforce: команда не найдена...
Аналогичная команда: 'getenforce'
[root@localhost masilkina]# getenforce
Permissive
[root@localhost masilkina]#
```

Figure 3.2: Отключение системы запретов

3.2 Создание программ

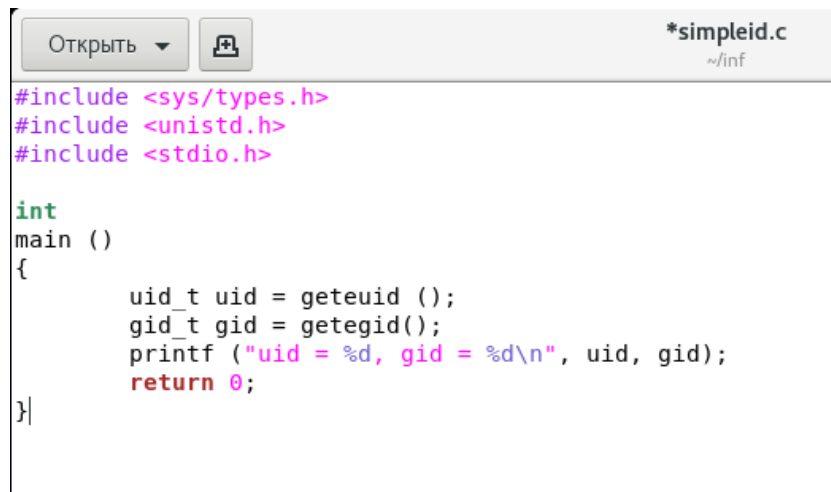
Я зашла в систему от имени пользователя guest, создала файл simpleid.c (рис 3. @fig:003)



```
guest@localhost:~/inf
Файл Правка Вид Поиск Терминал Справка
[guest@localhost ~]$ mkdir inf
[guest@localhost ~]$ cd inf
[guest@localhost inf]$ touch simpleid.c
[guest@localhost inf]$ gedit simpleid.c
```

Figure 3.3: Создание файла и его редактирование

Записала в файл simpleid.c требуемый код (рис 4. @fig:004)



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

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

Figure 3.4: Код программы

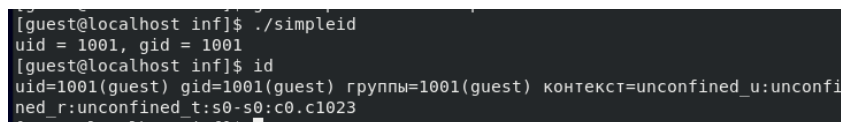
Скомпилировала файл simpleid.c (рис 5. @fig:005)



```
[guest@localhost inf]$ gcc simpleid.c -o simpleid
[guest@localhost inf]$
```

Figure 3.5: Компиляция

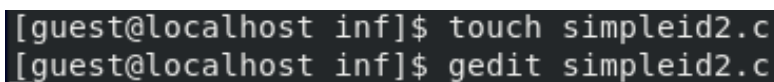
Запустила программу simpleid и системную программу id для сравнения полученных результатов, выведенна информация была идентичной (рис 6. @fig:006)



```
[guest@localhost inf]$ ./simpleid
uid = 1001, gid = 1001
[guest@localhost inf]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost inf]$
```

Figure 3.6: Сравнение вывода программ

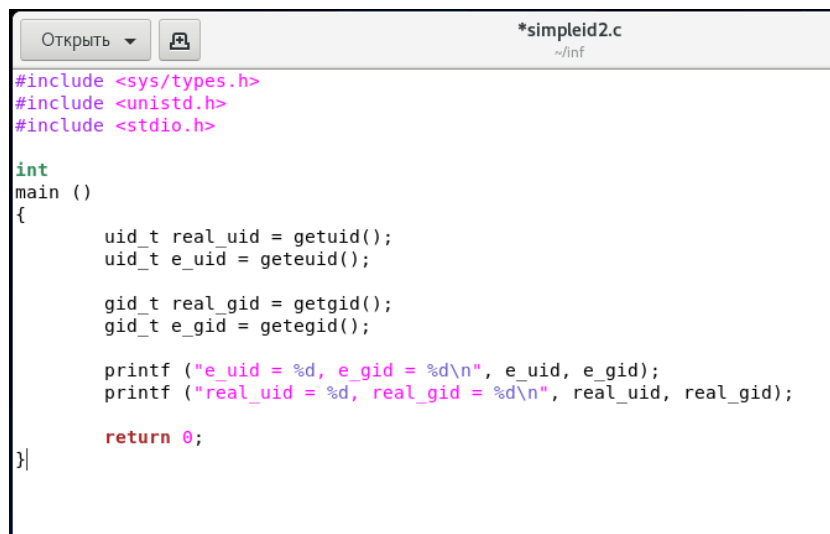
Далее я создала второй файл программы simpleid2.c (рис 7. @fig:007)



```
[guest@localhost inf]$ touch simpleid2.c
[guest@localhost inf]$ gedit simpleid2.c
```

Figure 3.7: Создание и редактирование файла

Записала в нее код из инструкции к лабораторной работе, он сложнее нежели прошлый, так как в него добавился вывод действительных идентификаторов (рис 8. @fig:008)



```
*simpleid2.c
~/inf

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

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

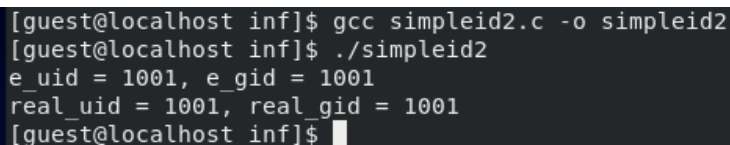
    gid_t real_gid = getgid();
    gid_t e_gid = getegid();

    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 3.8: Код программы

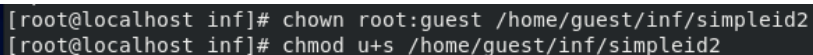
Скомпилировала и запустила данный файл. Вывод оказался аналогичен выводу предыдущей программы (рис 9. @fig:009)



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

Figure 3.9: Компиляция и вывод

От имени суперпользователя выполнила команды: для изменения владельца программы и права, с которыми пользователь может выполнить файл только с разрешением владельца (рис 10. @fig:010)



```
[root@localhost inf]# chown root:guest /home/guest/inf/simpleid2
[root@localhost inf]# chmod u+s /home/guest/inf/simpleid2
```

Figure 3.10: Выполнение команд

Запустила simpleid2 и системную программу id для сравнения полученных результатов, выведенная информация отличалась в одном пункте. Прodelала тоже самое относительно SetGID-бита (рис 11. @fig:011)

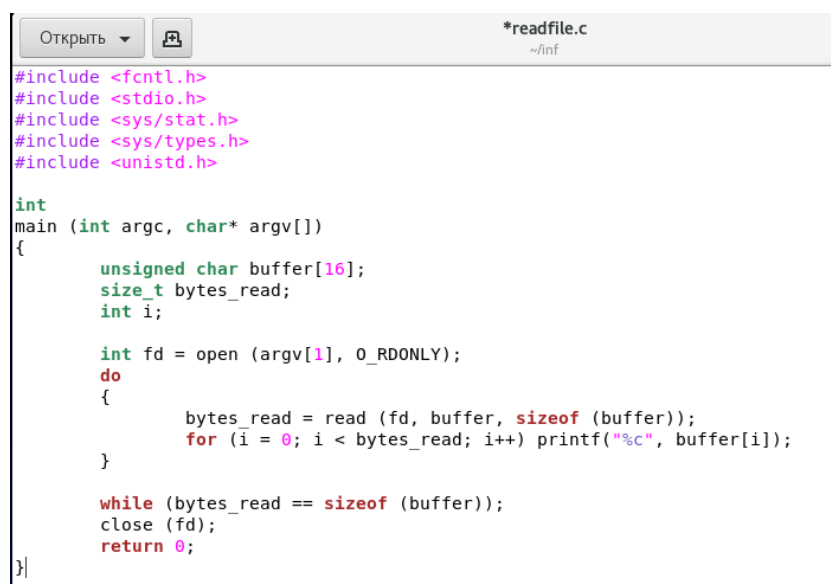
```
[guest@localhost inf]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 17648 ноя 12 12:47 simpleid2
[guest@localhost inf]$ ./simpleid2
e_uid = 0, e_gid = 1001
real_uid = 1001, real_gid = 1001
[guest@localhost inf]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost inf]$
```

Figure 3.11: Сравнение вывода программ

Создала новый файл readfile.c и записала в него код из инструкции, скомпилировала ее (рис 12. @fig:012) (рис 13. @fig:013)

```
[guest@localhost inf]$ touch readfile.c
[guest@localhost inf]$ gedit readfile.c
```

Figure 3.12: Создание файла



```
*readfile.c
~/inf

#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 3.13: Код программы

Сменила владельца файла и изменила права так, чтобы только суперпользователь (root)

мог прочитать его, а guest не мог и сделала проверку (рис 14. @fig:014) (рис 15. @fig:015) (рис 16. @fig:016)

```
[root@localhost inf]# chown root /home/guest/inf/readfile.c
```

Figure 3.14: Смен владельца

```
[root@localhost inf]# chmod 300 /home/guest/inf/readfile.c
```

Figure 3.15: Изменение прав

```
[guest@localhost inf]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@localhost inf]$
```

Figure 3.16: Проверка

Установила SetUID-бит и проверила (рис 17. @fig:017) (рис 18. @fig:018)

```
[root@localhost inf]# chown root /home/guest/inf/readfile
[root@localhost inf]# chmod u+s /home/guest/inf/readfile
```

Figure 3.17: Установка SetUID-бита

[illegible]

Figure 3.18: Проверка

3.3 Исследование Sticky-бита

Для начала я выяснила установлен ли атрибут Sticky на директории /tmp. Атрибут установлен и обозначается “t”. От имени пользователя guest я создала файл, в который записала слово

“test”, посмотрела атрибуты у файла, установила права для разрешения чтения и записи для категории пользователей “others” и проверила, что они верны (рис 19. @fig:019)

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 12 13:38 tmp
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 ноя 12 13:40 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 ноя 12 13:40 /tmp/file01.txt
[guest@localhost ~]$
```

Figure 3.19: Проверка наличия атрибута, создание файла, установка прав на него

Далее я зашла под пользователем guest2 и попыталась отредактировать дважды данный файл, заменив в нем слово. Данное действие было выполнено успешно. Мне также удавалось проверять содержимое файла, однако удалить файл не получилось (рис 20. @fig:020)

```
[guest@localhost ~]$ su - guest2
Пароль:
[guest2@localhost ~]$ cat /tmp/file01.txt
test
[guest2@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test2
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test3
[guest2@localhost ~]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
```

Figure 3.20: Выполнение действий

Следующим действием я сняла атрибут Sticky на директории /tmp от имени суперпользователя. (рис 21. @fig:021)

```
[guest2@localhost ~]$ su -
Пароль:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
ВЫХОД
```

Figure 3.21: Удаление атрибута

Дальше я попыталась повторить все ранее проделанные действия от пользователя guest2 и мне удалось перезаписать файл, прочитать и удалить (рис 22. @fig:022)

```
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 ноя 12 13:47 tmp
[guest2@localhost ~]$ echo "test2" > /tmp/file01.txt
[guest2@localhost ~]$ cat /tmp/file01.txt
test2
[guest2@localhost ~]$ rm /tmp/file01.txt
[guest2@localhost ~]$ ls /tmp
anaconda.log
dbus.log
dnf.librepo.log
ks-script-59fo_2iw
ks-script-msvkl4ep
packaging.log
program.log
sensitive-info.log
systemd-private-628a3a516c0d4fffb1e74fda0221c8ec-color.service-ET9DVf
systemd-private-628a3a516c0d4fffb1e74fda0221c8ec-fwupd.service-JnWQsf
systemd-private-628a3a516c0d4fffb1e74fda0221c8ec-ModemManager.service-9D3mDg
systemd-private-628a3a516c0d4fffb1e74fda0221c8ec-rtkit-daemon.service-t8FC0i
tracker-extract-files.1000
tracker-extract-files.1001
```

Figure 3.22: Выполнение действий без атрибута Sticky

Повысила права до суперпользователя и вернула атрибут t на директорию (рис 23. @fig:023)

```
[guest2@localhost ~]$ su -
Пароль:
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]# exit
ВЫХОД
[guest2@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 ноя 12 13:49 tmp
[guest2@localhost ~]$
```

Figure 3.23: Возвращение атрибута t

4 Выводы

При выполнении данной лабораторной работы я получила практические навыки работы в консоли с дополнительными атрибутами файлов. Я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

5 Библиография

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов.