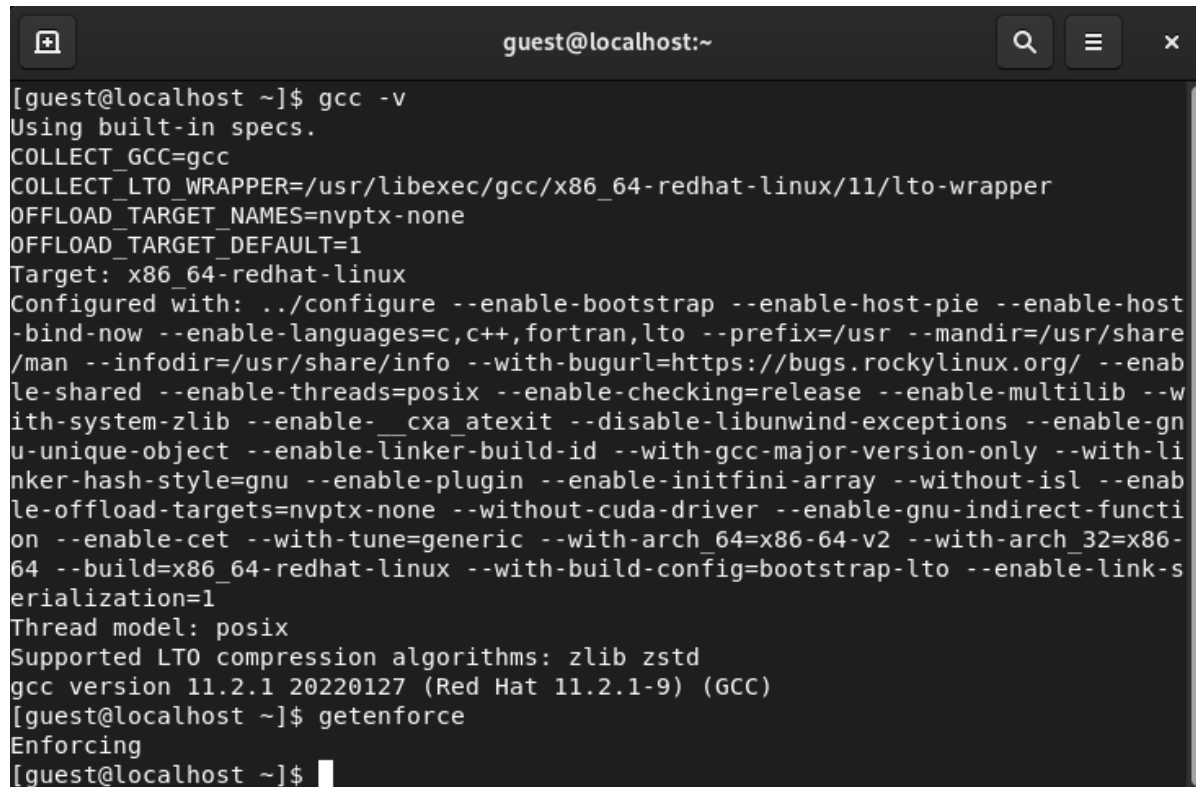


Цель работы

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.

Выполнение лабораторной работы

Проверю, установлен ли у меня компилятор gcc командой gcc -v.



```
guest@localhost:~  
[guest@localhost ~]$ gcc -v  
Using built-in specs.  
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  
Target: x86_64-redhat-linux  
Configured with: ../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/ --enab  
le-shared --enable-threads=posix --enable-checking=release --enable-multilib --w  
ith-system-zlib --enable-__cxa_atexit --disable-libunwind-exceptions --enable-gn  
u-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-li  
nker-hash-style=gnu --enable-plugin --enable-initfini-array --without-isl --enab  
le-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-functi  
on --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-lto --enable-link-s  
erialization=1  
Thread model: posix  
Supported LTO compression algorithms: zlib zstd  
gcc version 11.2.1 20220127 (Red Hat 11.2.1-9) (GCC)  
[guest@localhost ~]$ getenforce  
Enforcing  
[guest@localhost ~]$
```

Создание программы

Войду в систему от имени пользователя guest.

Создам программу simpleid.c.



```
Activities Text Editor Oct 13 14:01  
Open *simpleid.c Save  
1 #include <sys/types.h>  
2 #include <unistd.h>  
3 #include <stdio.h>  
4  
5 int main()  
6 {  
7     uid_t uid = geteuid();  
8     gid_t gid = getegid();  
9     printf("uid=%d, gid=%d\n", uid, gid);  
10    return 0;  
11 }
```

Скомпилирую программу командой gcc simpleid.c -o simpleid и удостоверюсь, что файл программы создан

Выполню программу simpleid командой ./simpleid

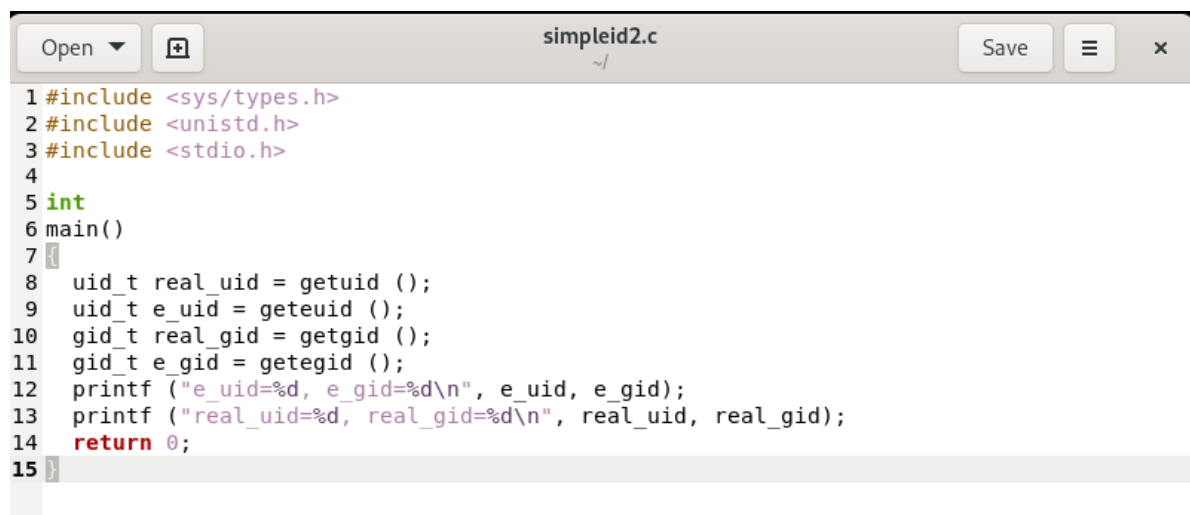
```
[guest@localhost ~]$ gedit ~/simpleid.c
[guest@localhost ~]$ gcc simpleid.c -o sipleid
[guest@localhost ~]$ ls
Desktop  Documents  Music      Public      sipleid     Videos
dir1     Downloads  Pictures   simpleid.c  Templates
[guest@localhost ~]$

[guest@localhost ~]$ ./sipleid
uid=1001, gid=1001
```

Выполню системную программу id командой id. Результат совпадает.

```
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$
```

Усложню программу, добавив вывод действительных идентификаторов. Создам новый файл simpleid2.c



```
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    gid_t real_gid = getgid ();
11    gid_t e_gid = getegid ();
12    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
13    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
14    return 0;
15 }
```

Скомпилирую и запущу simpleid2.c

```
[guest@localhost ~]$ gcc simpleid2.c -o simpleid2
[guest@localhost ~]$ ls
Desktop  Documents  Music      Public      simpleid2.c  sipleid     Videos
dir1     Downloads  Pictures   simpleid2   simpleid.c   Templates
[guest@localhost ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost ~]$
```

Работа с e SetUID-битом

От имени суперпользователя выполню команды:

chown root:guest /home/guest/simpleid2

chmod u+s /home/guest/simpleid2

```
[guest@localhost ~]$ su
Password:
```

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

Команда `chown root:guest /home/guest/simpleid2` меняет владельца файла. Команда `chmod u+s /home/guest/simpleid2` меняет права доступа к файлу.

Проверю правильность установки новых атрибутов и смены владельца файла `simpleid2` командой: `ls -l simpleid2`

```
[root@localhost guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 Oct 13 14:09 simpleid2
[root@localhost guest]#
```

Запущу `simpleid2` и `id`, команды: `./simpleid2` и `id`

```
[root@localhost guest]# ./simpleid2
e_uid=0, e_gid=0
real uid=0, real_gid=0
[root@localhost guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@localhost guest]#
```

После выполнения команд изменился параметр `e_uid`.

SetGID-бит

От имени суперпользователя выполняю команды:

`chmod u-s /home/guest/simpleid2` – чтобы отменить изменения на прошлом шаге

`chmod g+s /home/guest/simpleid2`

```
[root@localhost guest]# chmod u-s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]#
```

Проверю правильность установки новых атрибутов и смены владельца файла `simpleid2` командой: `ls -l simpleid2`

```
[root@localhost guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 26008 Oct 13 14:09 simpleid2
[root@localhost guest]#
```

Запущу `simpleid2` и `id`, команды: `./simpleid2` и `id`. Ничего не изменилось.

```
[root@localhost guest]# ./simpleid2
e_uid=0, e_gid=1001
real uid=0, real_gid=0
[root@localhost guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
[root@localhost guest]#
```

Создам программу `readfile.c`

```
Open ▾ + readfile.c ~/ Sav
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int
8 main (int argc, char* argv[])
9 {
10  unsigned char buffer[16];
11  size_t bytes_read;
12  int i;
13
14  int fd = open (argv[1], O_RDONLY);
15  do
16  {
17      bytes_read = read (fd, buffer, sizeof(buffer));
18      for(i=0; i < bytes_read; ++i) printf("%c", buffer[i]);
19  }
20
21  while(bytes_read == sizeof(buffer));
22  close(fd);
23  return 0;
24 }
```

Скомпилирую её командой: gcc readfile.c -o readfile

```
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$
```

Сменю владельца у файла readfile.c и изменю права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог.

```
[root@localhost guest]# chown root:guest /home/guest/readfile.c
[root@localhost guest]# chmod 700 /home/guest/readfile.c
[root@localhost guest]#
```

Проверю, что пользователь guest не может прочитать файл readfile.c.

```
[guest@localhost ~]$ ls -l readfile.c
-rwx----- 1 root guest 488 Oct 13 14:34 readfile.c
[guest@localhost ~]$ cat readfile.c
cat: readfile.c: Permission denied
[guest@localhost ~]$
```

Сменю у программы readfile владельца и установлю SetUID-бит.

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

Проверю, может ли программа readfile прочитать файл readfile.c

```
[guest@localhost ~]$ ./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;
}
```

Проверю, может ли программа readfile прочитать файл /etc/shadow

```
[guest@localhost ~]$ ./readfile /etc/shadow
root:$6$khu0fp6z9Wbi3Ab9$V352x7yB.YJeWa2PdCZs1gztonvR9mbsBeg..hAuzKb//x2p3S/xrfp
K3yE8MRfViJewt8n8JSbUvg8QTm9LG0::0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19252::::::
dbus:!!:19252::::::
polkitd:!!:19252::::::
rtkit:!!:19252::::::
sssd:!!:19252::::::
avahi:!!:19252::::::
pipewire:!!:19252::::::
```

Поскольку у программы установлен SetUID-бит, то ей временно предоставляются права владельца файла (суперпользователя). Поэтому программа может прочитать файл с правами доступа только для владельца суперпользователя.

Исследование Sticky-бита

Выясню, установлен ли атрибут Sticky на директории /tmp, для чего выполню команду `ls -l / | grep tmp`

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct 13 14:38 tmp
[guest@localhost ~]$
```

От имени пользователя guest создам файл file01.txt в директории /tmp со словом test:
`echo "test" > /tmp/file01.txt`

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test
```

Просмотрю атрибуты у только что созданного файла и разрешу чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```

```
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 Oct 13 14:40 /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 Oct 13 14:40 /tmp/file01.txt
[guest@localhost ~]$
```

От пользователя guest2 (не являющегося владельцем) попробую прочитать файл /tmp/file01.txt: cat /tmp/file01.txt

```
[guest@localhost ~]$ cat /tmp/file01.txt
test
[guest@localhost ~]$
```

От пользователя guest2 попробую дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" >> /tmp/file01.txt

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt

```
[guest@localhost ~]$ echo "test2" >> /tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test
test2
```

От пользователя guest2 попробую записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt

```
[guest@localhost ~]$ cat /tmp/file01.txt
test3
```

От пользователя guest2 попробую удалить файл /tmp/file01.txt командой rm /tmp/file01.txt

```
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: cannot remove '/tmp/file01.txt': Operation not permitted
```

Мне не удалось удалить файл.

Повышу свои права до суперпользователя следующей командой su и выполню после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp

```
[guest2@localhost guest]$ su
Password:
[root@localhost guest]# chmod -t /tmp
```

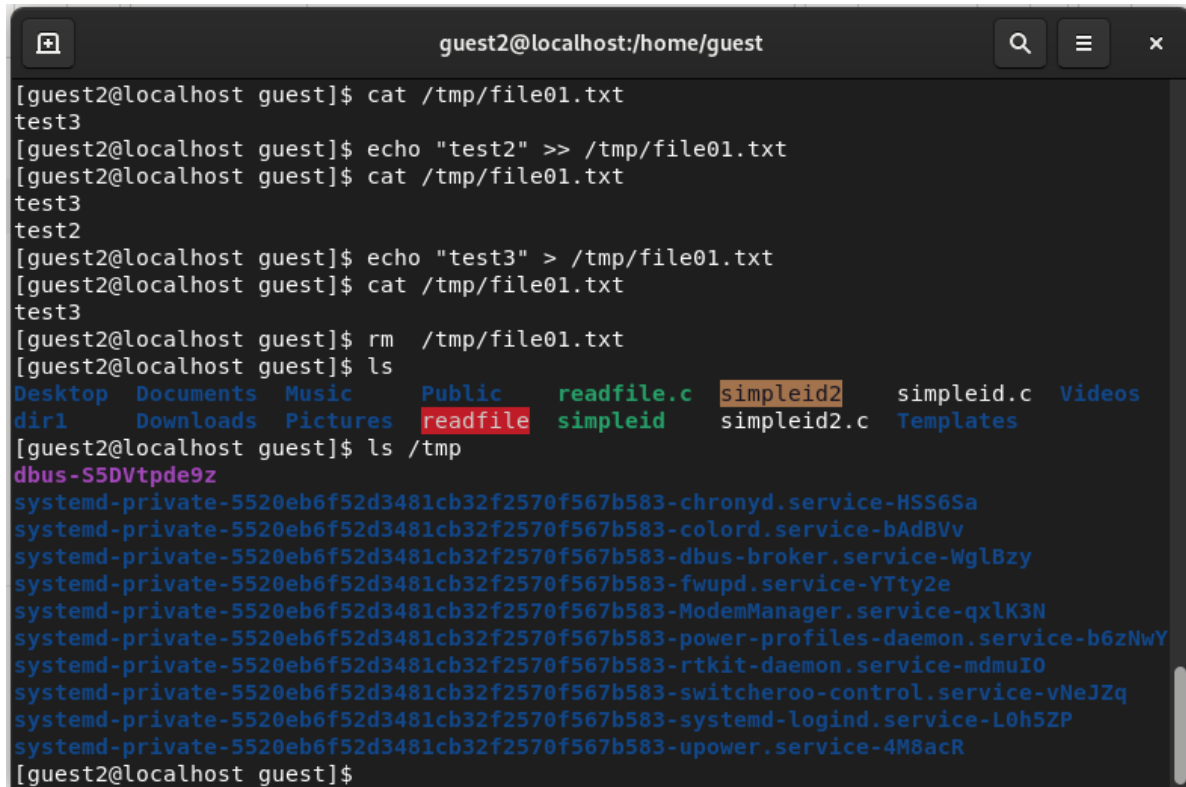
Покину режим суперпользователя командой exit

```
[root@localhost guest]# exit
exit
[guest2@localhost guest]$
```

От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp

```
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 Oct  4 01:38 tmp
```

Повторю предыдущие шаги.

A screenshot of a terminal window titled 'guest2@localhost:/home/guest'. The terminal shows a series of commands and their outputs. First, 'cat /tmp/file01.txt' outputs 'test3'. Then, 'echo "test2" >> /tmp/file01.txt' is executed. Next, 'cat /tmp/file01.txt' outputs 'test3' followed by 'test2'. Then, 'echo "test3" > /tmp/file01.txt' is executed. Next, 'cat /tmp/file01.txt' outputs 'test3'. Then, 'rm /tmp/file01.txt' is executed. Finally, 'ls' is executed, showing a list of files and directories including Desktop, Documents, Music, Public, readfile.c, simpleid2, simpleid.c, Videos, dir1, Downloads, Pictures, readfile, simpleid, simpleid2.c, and Templates. Then, 'ls /tmp' is executed, showing a list of systemd-private directories and services. The terminal window has a search icon, a menu icon, and a close icon in the top right corner.

```
guest2@localhost:/home/guest
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
[guest2@localhost guest]$ ls
Desktop  Documents  Music      Public    readfile.c  simpleid2  simpleid.c  Videos
dir1     Downloads  Pictures   readfile  simpleid    simpleid2.c  Templates
[guest2@localhost guest]$ ls /tmp
dbus-S5DVtpde9z
systemd-private-5520eb6f52d3481cb32f2570f567b583-chronyd.service-HSS6Sa
systemd-private-5520eb6f52d3481cb32f2570f567b583-colord.service-bAdBVv
systemd-private-5520eb6f52d3481cb32f2570f567b583-dbus-broker.service-WglBzy
systemd-private-5520eb6f52d3481cb32f2570f567b583-fwupd.service-YTty2e
systemd-private-5520eb6f52d3481cb32f2570f567b583-ModemManager.service-qxLK3N
systemd-private-5520eb6f52d3481cb32f2570f567b583-power-profiles-daemon.service-b6zNwY
systemd-private-5520eb6f52d3481cb32f2570f567b583-rtkit-daemon.service-mdmuIO
systemd-private-5520eb6f52d3481cb32f2570f567b583-switcheroo-control.service-vNeJZq
systemd-private-5520eb6f52d3481cb32f2570f567b583-systemd-logind.service-L0h5ZP
systemd-private-5520eb6f52d3481cb32f2570f567b583-upower.service-4M8acR
[guest2@localhost guest]$
```

Мне удалось удалить файл от имени пользователя, не являющегося его владельцем.

Это связано с тем, что Sticky-bit позволяет защищать файлы от случайного удаления, когда несколько пользователей имеют права на запись в один и тот же каталог. Если

у файла атрибут t стоит, значит пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. Если же этот атрибут не установлен, то удалить файл могут все пользователи, которым позволено удалять файлы из каталога.

Повышу свои права до суперпользователя и верну атрибут t на директорию /tmp:

su

chmod +t /tmp

exit

```
[guest2@localhost guest]$ su
Password:
[root@localhost guest]# chmod +t /tmp
[root@localhost guest]# ls -l / | grep tmp
drwxrwxrwt. 17 root root 4096 Oct  4 01:40 tmp
[root@localhost guest]# exit
exit
[guest2@localhost guest]$
```


Вывод

В ходе данной лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID-, SetGID- и Sticky-битов. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

- [Кулябов Д. С., Королькова А. В., Геворкян М. Н. Лабораторная работа №5](#)