

# 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.

```
[raaldorikhim@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)
[raaldorikhim@localhost ~]$
```

## Создание программы

Войду в систему от имени пользователя guest.



Создам программу simpleid.c.



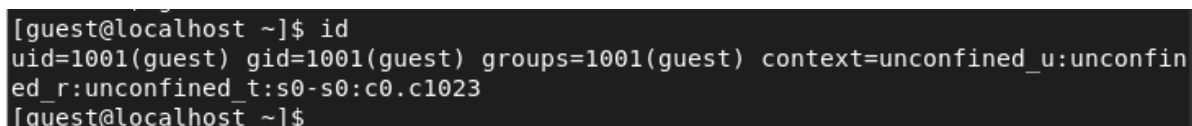
Скомпилирую программу командой `gcc simpleid.c -o simpleid` и удостоверюсь, что файл программы создан

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

Выполню программу simpleid командой `./simpleid`



Выполню системную программу id командой `id`. Результат совпадает.



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

```
Open  + simpleid2.c  Save  ⋮  ×

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  simpleid.c  Videos
dir1     Downloads  Pictures   simpleid    simpleid2.c  Templates
[guest@localhost ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
```

### 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]#
```

```
[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
```

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  4 01:05 simpleid2
```

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
```

Проверю правильность установки новых атрибутов и смены владельца файла

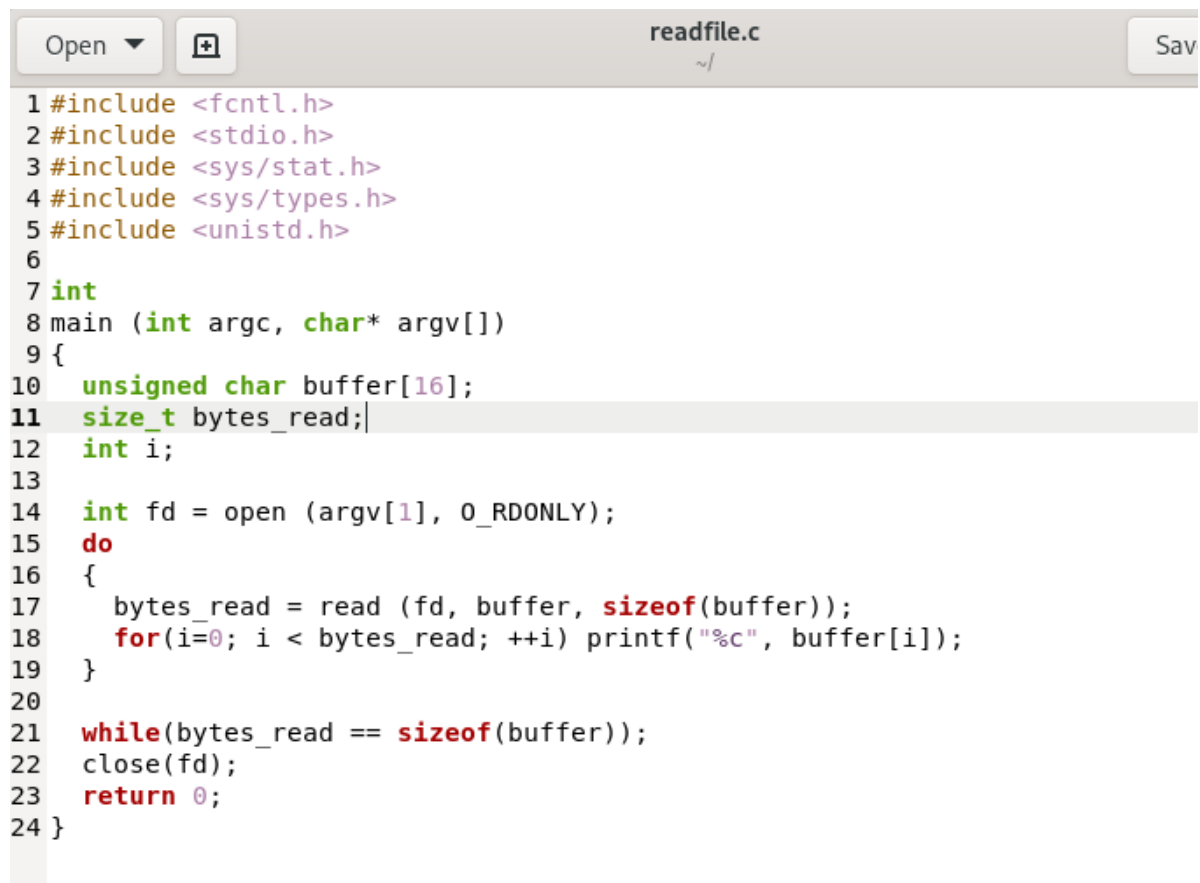
simpleid2 командой: ls -l simpleid2

```
[root@localhost guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 26008 Oct  4 01:05 simpleid2
```

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
```

Создам программу readfile.c



```
Open  +  readfile.c  Save
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
```

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

```
guest@localhost:~
[guest@localhost ~]$ ls -l readfile.c
-rwx-----. 1 root guest 431 Oct  4 01:23 readfile.c
[guest@localhost ~]$ cat readfile.c
cat: readfile.c: Permission denied
```

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
```

Проверю, может ли программа readfile прочитать файл readfile.c

```
guest@localhost:~
[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:$6$QBNs9xzQmhd8W.DN$cRihXrj2e20Nl/HVQdbFG3tXQFbRl0SuEBMQKl2stFfGjhQ03r11BsR3AGPy
M695vNBtz6agb4p4PNLzI7YJj...: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:!:19243::::::
dbus:!:19243::::::
polkitd:!:19243::::::
rtkit:!:19243::::::
sssd:!:19243::::::
avahi:!:19243::::::
pipewire:!:19243::::::
libstoragemgmt:!:19243::::::
tss:!:19243::::::
geoclue:!:19243::::::
cockpit-ws:!:19243::::::
```

```
cockpit-wsinstance:!:19243::::::
setroubleshoot:!:19243::::::
flatpak:!:19243::::::
colord:!:19243::::::
clevis:!:19243::::::
gdm:!:19243::::::
systemd-oom:!*:19243::::::
pesign:!:19243::::::
gnome-initial-setup:!:19243::::::
sshd:!:19243::::::
chrony:!:19243::::::
dnsmasq:!:19243::::::
tcpdump:!:19243::::::
raaldorikhim:$6$8nSPhVGR06Vd6A9h$JWhIq0GSNoe/DcpgqWuwVQDw64krosu3lAJukQemHXjDNBpkG2uY
U0nRzZu6SGFZuNIgBhMIzGc4oqCx7E75h0::0:99999:7:::
guest:$6$I.Jp0v8olosXqDjz$f3WtDo5Jvk7wCj8puoWX6maKZ8gAbVREzzc.WTsPge0QgyepjglmFxaCJKZ
Zw03iwm2Rn4wVvNG8sdh4UR/tt1:19248:0:99999:7:::
guest2:$6$KeZb.EEWkIiNCxU1$QLUX95AiVRTm9VkwpxFZZ52Slcme6xkwFhF5ZPYDSZVqmV2xlzJ0Ke7ZVR
4FzGE/8kEeqM32nvM3ZaUpuFZqr/:19258:0:99999:7:::
```

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. 16 root root 4096 Oct  4 01:28 tmp
```

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
```

Посмотрю 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  4 01:32 /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  4 01:32 /tmp/file01.txt
```

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

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test
```

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

```
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$
```

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt

```
[guest2@localhost guest]$ 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

```
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$
```

Мне удалось выполнить операцию.

Проверю содержимое файла командой cat /tmp/file01.txt

```
[guest2@localhost guest]$ 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
```



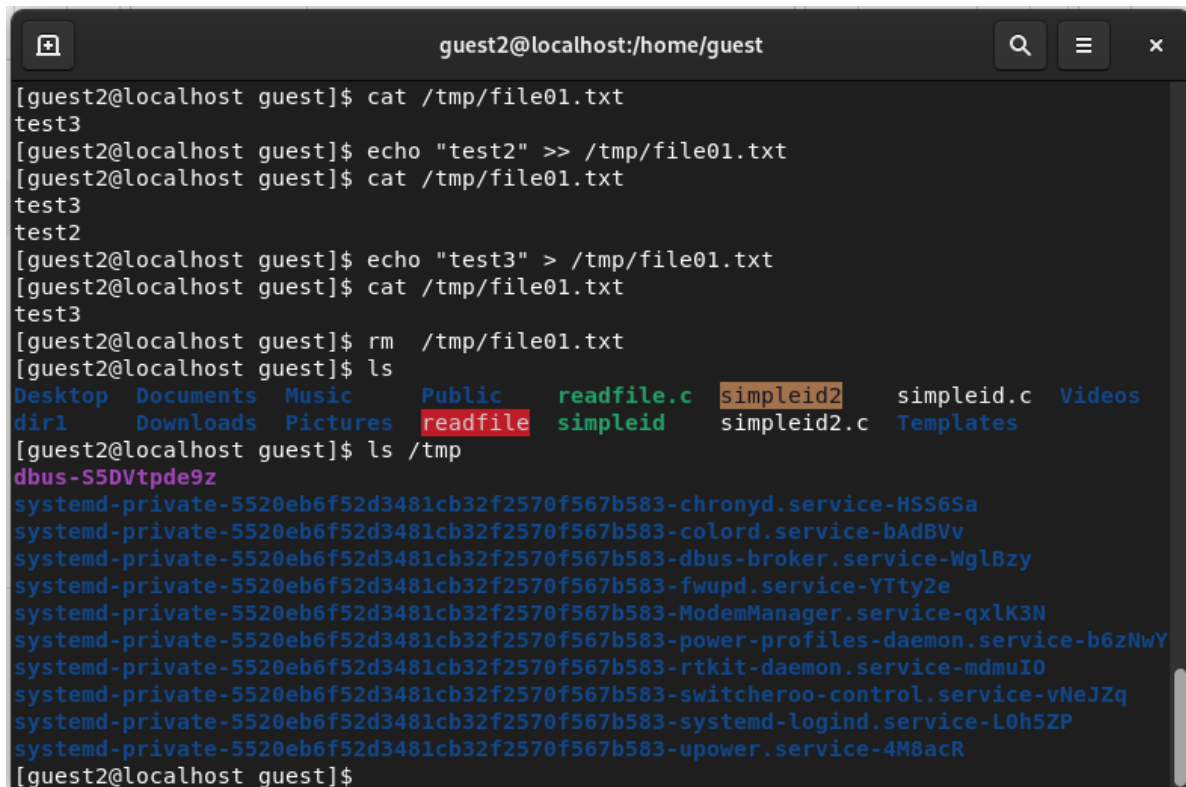
Покину режим суперпользователя командой exit

```
[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 terminal window titled 'guest2@localhost:/home/guest' with search, menu, and close icons in the title bar. The terminal shows a series of commands and their outputs. First, 'cat /tmp/file01.txt' outputs 'test3'. Then, 'echo "test2" >> /tmp/file01.txt' is followed by 'cat /tmp/file01.txt' which outputs 'test3' and 'test2'. Next, 'echo "test3" > /tmp/file01.txt' is followed by 'cat /tmp/file01.txt' which outputs 'test3'. Then, 'rm /tmp/file01.txt' is executed. Finally, 'ls' shows a directory listing with files like 'Desktop', 'Documents', 'Music', 'Public', 'readfile.c', 'simpleid2', 'simpleid.c', 'Videos', 'dir1', 'Downloads', 'Pictures', 'readfile', 'simpleid', 'simpleid2.c', and 'Templates'. Then, 'ls /tmp' shows a long list of systemd-private directories. The prompt is '[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](#)