

Лабораторная работа №5

Андрианова Марина Георгиевна
RUDN University, Moscow,
Russian Federation
2024, 04 October

Цель работы

Целью данной работы является 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

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc. Также осуществляется отключение системы запретом (рис.1).

```
[mgandrianova@localhost ~]$ gcc -v
Используются внутренние спецификации.
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
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../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/ --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-gcc-major-version-only --enable-plugin --enable-initfini-array --without-isl --enable-multilib --with-linker-hash-style=gnu --enable-offload-targets=nvptx-none --without-cuda-driver --enable-gnu-indirect-function --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-serialization=1
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.4.1 20231218 (Red Hat 11.4.1-3) (GCC)
[mgandrianova@localhost ~]$ sudo setenforce 0
[sudo] пароль для mgandrianova:
[mgandrianova@localhost ~]$ getenforce
Permissive
[mgandrianova@localhost ~]$
```

Рис.1. Подготовка к лабораторной работе

Осуществляется вход от имени пользователя guest (рис. 2).

```
[mgandrianova@localhost ~]$ su guest  
Пароль:  
[guest@localhost mgandrianova]$ touch simpleid.c
```

Рис.2. Вход от имени пользователя guest

Создание файла simpleid.c (рис. 3)

```
[guest@localhost ~]$ touch simpleid.c  
[guest@localhost ~]$ nano simpleid.c
```

Рис.3. Создание файла simpleid.c

Содержимое файла выглядит следующим образом (рис. 4)



```
GNU nano 5.6.1 simpleid.c
#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;
}
```

Рис.4. Содержимое файла

Компилирую файл, проверяю, что он скомпилировался (рис. 5)

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ls
simpleid      Видео      Загрузки    Музыка      'Рабочий стол'
simpleid.c    Документы  Изображения  Общедоступные  Шаблоны
[guest@localhost ~]$
```

Рис.5. Компиляция файла

Запускаю исполняемый файл (рис. 6)

```
[guest@localhost ~]$ ./simpleid
uid=1001, gid=1001
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest),10(wheel) контекст=unconfined
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис.6. Сравнение команд

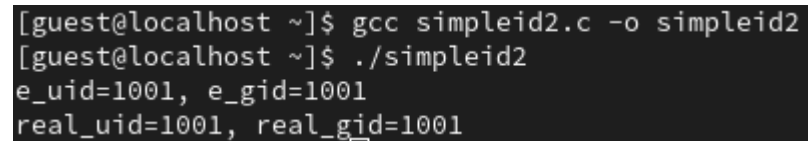
Создание файла simpleid2.c и запись в файл кода (рис. 7)



```
GNU nano 5.6.1 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;
}
```

Рис.7. Содержимое файла

Компиляция файла simpleid2.c. Запуск программы (рис. 8)



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

Рис.8. Компиляция файла simpleid2.c

С помощью `chown` изменяю владельца файла на суперпользователя, с помощью `chmod` изменяю права доступа (рис. 9)

```
[guest@localhost ~]$ sudo chown root:guest /home/guest/simpleid2
[sudo] пароль для guest:
[guest@localhost ~]$ sudo chmod u+s /home/guest/simpleid2
[guest@localhost ~]$ sudo ls -l /home/guest/simpleid2
-rwsr-xr-x. 1 root guest 17720 окт  4 20:52 /home/guest/simpleid2
```

Рис.9. Смена владельца файла и прав доступа к файлу

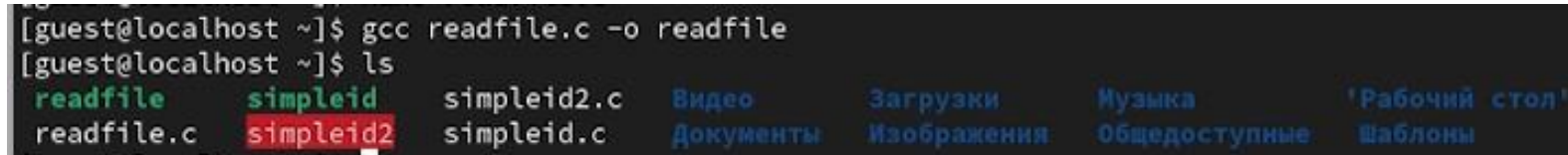
Создание файла readfile.c и запись кода в файл (рис. 10)



```
guest@localhost:~ — nano readfile.c
GNU nano 5.6.1 readfile.c
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;
}
```

Рис.10. Содержимое файла readfile.c

Компиляция файла readfile.c. Запуск программы (рис. 11)



```
[guest@localhost ~]$ gcc readfile.c -o readfile
[guest@localhost ~]$ ls
readfile  simpleid  simpleid2.c  Видео  Загрузки  Музыка  'Рабочий стол'
readfile.c simpleid2 simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
```

Рис.11. Компиляция файла readfile.c

Снова от имени суперпользователя меняю владельца файла `readfile`. Далее меняю права доступа так, чтобы пользователь `guest` не смог прочесть содержимое файла (рис. 12)

```
[guest@localhost ~]$ sudo chown root:guest /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod 700 /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod -r /home/guest/readfile.c
[guest@localhost ~]$ sudo chmod u+s /home/guest/readfile.c
[guest@localhost ~]$
```

Рис.12. Смена владельца файла и прав доступа к файлу

Проверка прочесть файл от имени пользователя `guest`. Прочесть файл не удастся (рис. 13)

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

Рис.13. Попытка прочесть содержимое файла

Пробуем прочесть файл от имени суперпользователя и чтение проходит успешно (рис. 14)

```
[guest@localhost ~]$ sudo /home/guest/readfile /etc/shadow
root:$6$CULeLnoUeBmXzEc.$uCzdaZzZdhgVq2g0pFCsMrNfZhL6cETw.gtPGDNpzlxhVH95HXCeeTDYlDZiRqQ8L.EqhGIwmClHGOf1dm12i.
::0:99999:7:::
bin:*:19820:0:99999:7:::
daemon:*:19820:0:99999:7:::
adm:*:19820:0:99999:7:::
lp:*:19820:0:99999:7:::
sync:*:19820:0:99999:7:::
shutdown:*:19820:0:99999:7:::
halt:*:19820:0:99999:7:::
mail:*:19820:0:99999:7:::
operator:*:19820:0:99999:7:::
games:*:19820:0:99999:7:::
ftp:*:19820:0:99999:7:::
nobody:*:19820:0:99999:7:::
```

Рис.14. Чтение файла от имени суперпользователя

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 15)

```
[guest@localhost ~]$ ls -l / | grep tmp  
drwxrwxrwt. 20 root root 4096 окт  4 21:05 tmp
```

Рис.15. Проверка атрибутов директории tmp

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 16)

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt  
[guest@localhost ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 окт  4 21:06 /tmp/file01.txt  
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt  
[guest@localhost ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 окт  4 21:06 /tmp/file01.txt
```

Рис.16. Создание файла, изменение прав доступа

Далее пробуем удалить файл, получаем отказ (рис. 17)

```
[guest@localhost ~]$ su guest2
Пароль:
[guest2@localhost guest]$ cat /tmp/file01.txt
test
```

Рис.17. Попытка удалить файл

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 18)

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

Рис.18. Смена атрибутов файла

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита удаление файла прошло успешно (рис. 19)

```
[guest2@localhost guest]$ echo "test" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
```

Рис.19. Попытка удалить файл

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 20)

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

Рис.20. Изменение атрибутов

Выводы

- Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты.
- Получила практические навыки работы в консоли с дополнительными атрибутами.
- Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.