

Отчет по лабораторной работе №5

Информационная безопасность

Астафьева Анна Андреевна НПИбд-01-18

Содержание

1	Цель работы	4
2	Теоретическое описание	5
3	Выполнение лабораторной работы	6
4	Выводы	15

List of Figures

3.1	Создание файла simpleid.c	6
3.2	Написание программы simpleid.c	7
3.3	Компиляция, выполнение программы	8
3.4	Создание программы simpleid2.c	9
3.5	Компиляция, выполнение программы	9
3.6	Выполнение	10
3.7	Выполнение	10
3.8	Создание и компиляция программы readfile.c	10
3.9	Изменение владельца и прав	10
3.10	Проверка	11
3.11	Изменение для программы readfile	11
3.12	Проверка	11
3.13	Проверка	12
3.14	Выполнение	12
3.15	Выполнение и проверка от пользователя guest2	13
3.16	Снятие атрибута “t” с директории /tmp	13
3.17	Проверка	13
3.18	Добавление атрибута “t” на директорию /tmp	14

1 Цель работы

Изучить механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получить практические навыки работы в консоли с дополнительными атрибутами. Рассмотреть работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

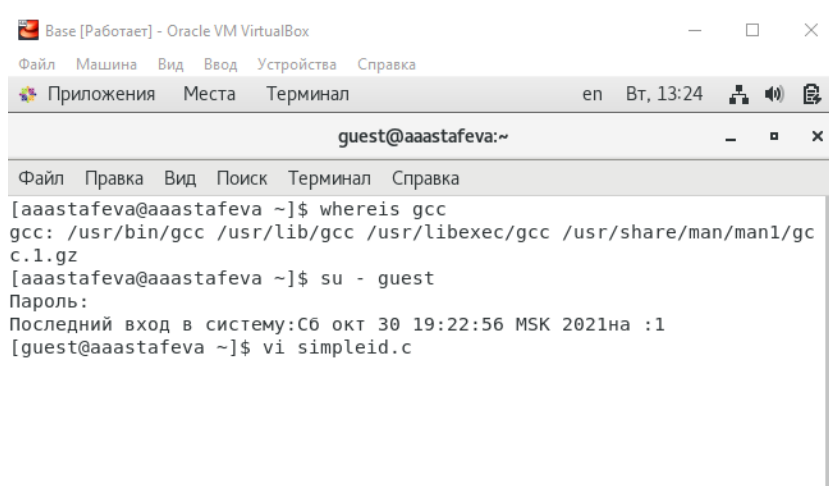
2 Теоретическое описание

В Linux, как и в любой многопользовательской системе, абсолютно естественным образом возникает задача разграничения доступа субъектов — пользователей к объектам — файлам дерева каталогов.

Setuid, Setgid и Sticky Bit - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги. Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Принцип работы Setgid очень похож на setuid с отличием, что файл будет запускаться пользователем от имени группы, которая владеет файлом. Последний специальный бит разрешения – это Sticky Bit . В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем.

3 Выполнение лабораторной работы

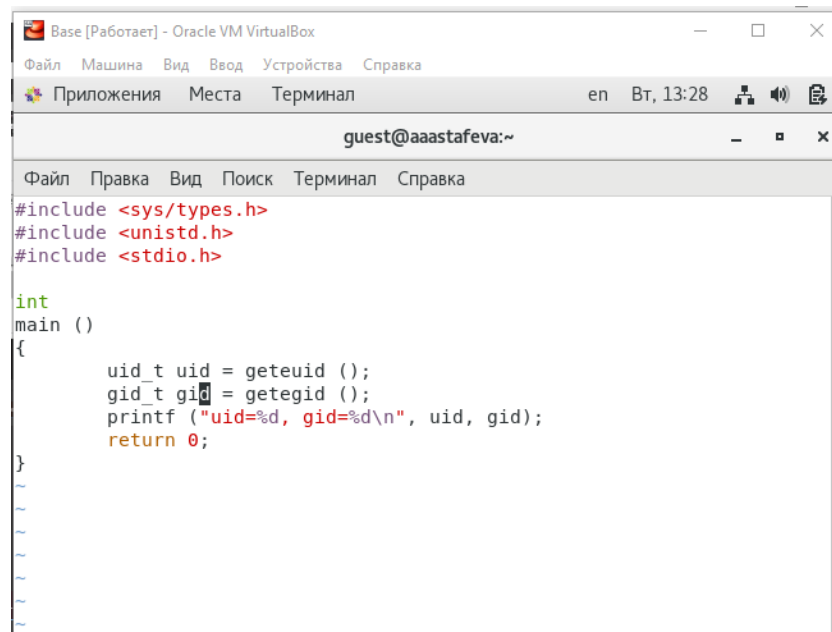
1. Вошла в систему от имени пользователя guest, создала программу simpleid.c.
(рис. 3.1), (рис. 3.2).



The screenshot shows a terminal window titled "Base [Работает] - Oracle VM VirtualBox". The window has a menu bar with "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". Below the menu bar is a toolbar with icons for "Приложения", "Места", "Терминал", and a language dropdown set to "en". The terminal prompt is "guest@aaastafeva:~". The terminal output shows the following commands and their results:

```
[aaastafeva@aaastafeva ~]$ whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz
[aaastafeva@aaastafeva ~]$ su - guest
Пароль:
Последний вход в систему:Сб окт 30 19:22:56 MSK 2021на :1
[guest@aaastafeva ~]$ vi simpleid.c
```

Figure 3.1: Создание файла simpleid.c



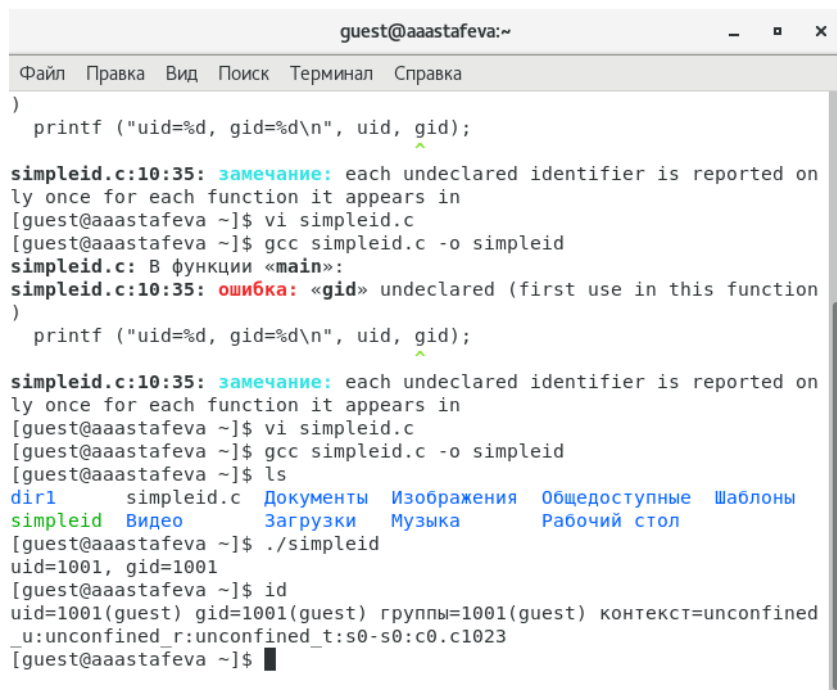
The screenshot shows a terminal window titled "Base [Работает] - Oracle VM VirtualBox". The window has a menu bar with "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". Below the menu bar is a toolbar with icons for applications, locations, and terminal. The terminal prompt is "guest@aaastafeva:~". The code being written is as follows:

```
#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 3.2: Написание программы simpleid.c

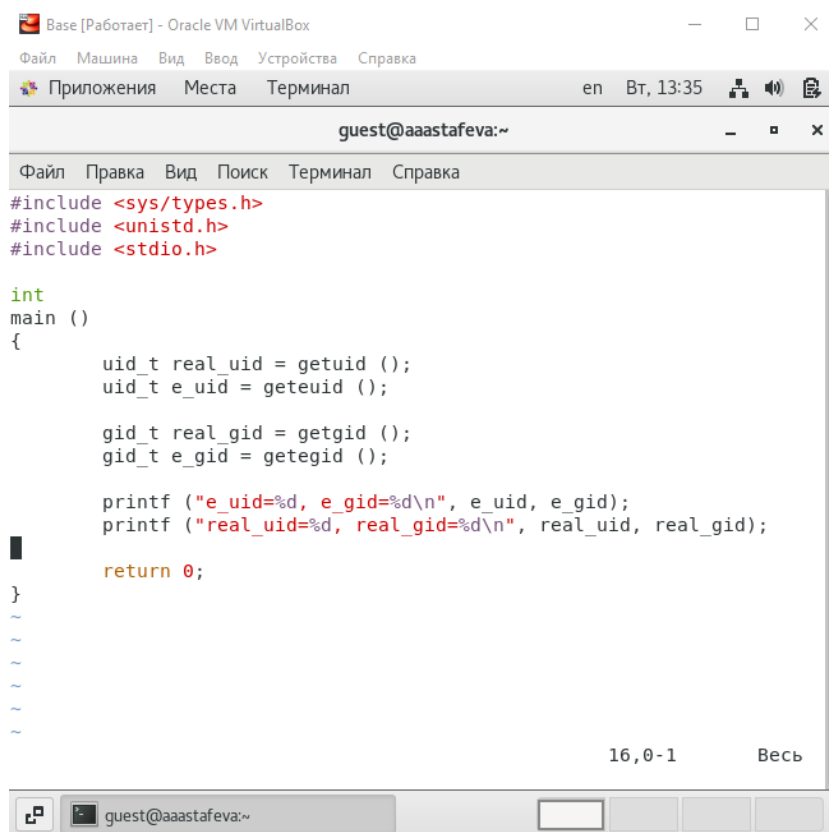
2. Скомпилировала программу и убедилась, что файл программы создан: gcc simpleid.c -o simpleid. Выполнила программу simpleid: ./simpleid. Выполнила системную программу id. И сравнила полученный результат с данными предыдущего пункта задания. (Данные одинаковы)(рис. 3.3).



```
guest@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
)  
    printf ("uid=%d, gid=%d\n", uid, gid);  
simpleid.c:10:35: замечание: each undeclared identifier is reported on  
ly once for each function it appears in  
[guest@aaastafeva ~]$ vi simpleid.c  
[guest@aaastafeva ~]$ gcc simpleid.c -o simpleid  
simpleid.c: В функции «main»:  
simpleid.c:10:35: ошибка: «gid» undeclared (first use in this function  
)  
    printf ("uid=%d, gid=%d\n", uid, gid);  
simpleid.c:10:35: замечание: each undeclared identifier is reported on  
ly once for each function it appears in  
[guest@aaastafeva ~]$ vi simpleid.c  
[guest@aaastafeva ~]$ gcc simpleid.c -o simpleid  
[guest@aaastafeva ~]$ ls  
dir1 simpleid.c Документы Изображения Общедоступные Шаблоны  
simpleid Видео Загрузки Музыка Рабочий стол  
[guest@aaastafeva ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@aaastafeva ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined  
_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@aaastafeva ~]$
```

Figure 3.3: Компиляция, выполнение программы

3. Усложнила программу, добавив вывод действительных идентификаторов.
(рис. 3.4).



The screenshot shows a terminal window titled "Base [Работает] - Oracle VM VirtualBox". The window has a menu bar with "Файл", "Машина", "Вид", "Ввод", "Устройства", and "Справка". Below the menu bar is a toolbar with icons for "Приложения", "Места", "Терминал", and "en". The terminal prompt is "guest@aaastafeva:~". The code being entered is as follows:

```
#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 ();

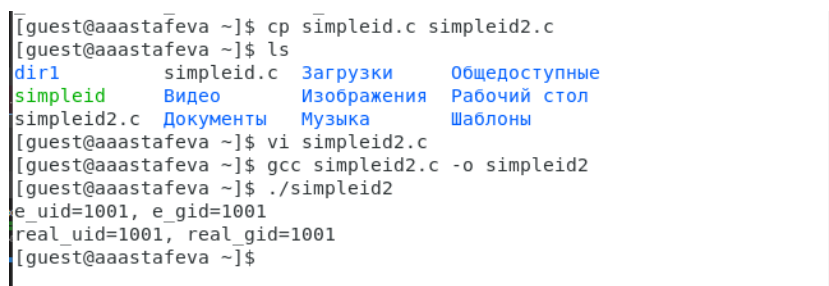
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);

    return 0;
}
```

At the bottom of the terminal window, the text "16,0-1" and "Весь" are visible.

Figure 3.4: Создание программы simpleid2.c

4. Скомпилировала и запустила simpleid2.c: `gcc simpleid2.c -o simpleid2; ./simpleid2` (рис. 3.5).



The screenshot shows a terminal window with the following commands and output:

```
[guest@aaastafeva ~]$ cp simpleid.c simpleid2.c
[guest@aaastafeva ~]$ ls
dir1          simpleid.c    Загрузки     Общедоступные
simpleid       Видео         Изображения  Рабочий стол
simpleid2.c    Документы    Музыка       Шаблоны
[guest@aaastafeva ~]$ vi simpleid2.c
[guest@aaastafeva ~]$ gcc simpleid2.c -o simpleid2
[guest@aaastafeva ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@aaastafeva ~]$
```

Figure 3.5: Компиляция, выполнение программы

5. От имени суперпользователя выполнила команды: `chown root:guest /home/guest/simpleid2; chmod u+s /home/guest/simpleid2`. (рис. 3.6).

```
root@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
[aaastafeva@aaastafeva ~]$ sudo -i  
[sudo] пароль для aaastafeva:  
[root@aaastafeva ~]# chown root:guest /home/guest/simpleid2  
[root@aaastafeva ~]# chmod u+s /home/guest/simpleid2  
[root@aaastafeva ~]#
```

Figure 3.6: Выполнение

С помощью первой команды для файла simpleid2 мы поменяли пользователя и группу на root и guest соответственно. С помощью второй установили разрешение для владельца на выполнение с разрешением суперпользователя.

6. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2: `ls -l simpleid2`. Запустила simpleid2 и `id`. (рис. 3.7).

```
[guest@aaastafeva ~]$ ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8576 ноя  9 13:35 simpleid2  
[guest@aaastafeva ~]$ ./simpleid2  
e_uid=0, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@aaastafeva ~]$ id  
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined  
u:unconfined r:unconfined t:s0-s0:c0.c1023  
[guest@aaastafeva ~]$
```

Figure 3.7: Выполнение

7. Создала и откомпилировала программу readfile.c: (рис. 3.8).

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

Figure 3.8: Создание и компиляция программы readfile.c

8. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог. (рис. 3.9).

```
[root@aaastafeva ~]# chown root /home/guest/readfile.c  
[root@aaastafeva ~]# chmod u+x /home/guest/readfile.c  
[root@aaastafeva ~]# chmod g-rw /home/guest/readfile.c  
[root@aaastafeva ~]# chmod o-r /home/guest/readfile.c  
[root@aaastafeva ~]#
```

Figure 3.9: Изменение владельца и прав

9. Проверила, что пользователь guest не может прочитать файл readfile.c. (рис. 3.10).

```
[guest@aaastafeva ~]$ ls -l readfile.c
-rw-rw-r--. 1 guest guest 420 ноя  9 13:57 readfile.c
[guest@aaastafeva ~]$ ls -l readfile.c
-rw-rw-r--. 1 root guest 420 ноя  9 13:57 readfile.c
[guest@aaastafeva ~]$ ls -l readfile.c
-rwx-----. 1 root guest 420 ноя  9 13:57 readfile.c
[guest@aaastafeva ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@aaastafeva ~]$
```

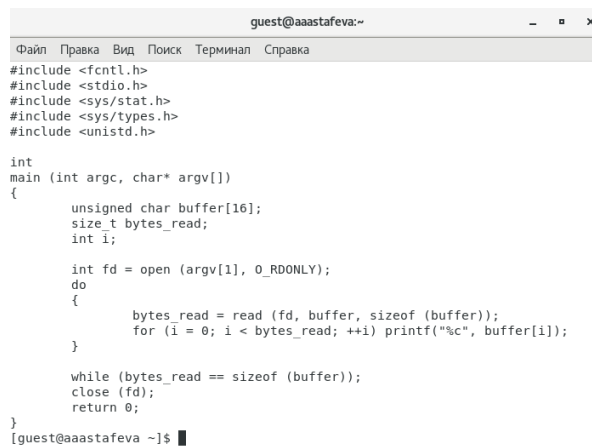
Figure 3.10: Проверка

10. Сменила у программы readfile владельца и установила SetU'D-бит. (рис. 3.11).

```
[root@aaastafeva ~]# chown root /home/guest/readfile
[root@aaastafeva ~]# chmod u+s /home/guest/readfile
```

Figure 3.11: Изменение для программы readfile

11. Проверила, может ли программа readfile прочитать файл readfile.c (может), проверила, может ли программа readfile прочитать файл /etc/shadow (может). (рис. 3.12). (рис. 3.13).



```
guest@aaastafeva:~
Файл Правка Вид Поиск Терминал Справка
#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@aaastafeva ~]$
```

Figure 3.12: Проверка


```
guest2@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
[aaastafeva@aaastafeva ~]$ su - guest2  
Пароль:  
Последний вход в систему:Сб окт 16 17:09:24 MSK 2021на pts/1  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test  
[guest2@aaastafeva ~]$ echo "test2" > /tmp/file01.txt  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test2  
[guest2@aaastafeva ~]$ echo "test" > /tmp/file01.txt  
[guest2@aaastafeva ~]$ echo "test2" >> /tmp/file01.txt  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test  
test2  
[guest2@aaastafeva ~]$ echo "test3" > /tmp/file01.txt  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test3  
[guest2@aaastafeva ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@aaastafeva ~]$
```

Figure 3.15: Выполнение и проверка от пользователя guest2

14. От суперпользователя выполнила команду, снимающую атрибут t (Sticky-бит) с директории /tmp: `chmod -t /tmp`. (рис. 3.16).

```
root@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
[root@aaastafeva ~]# chmod -t /tmp
```

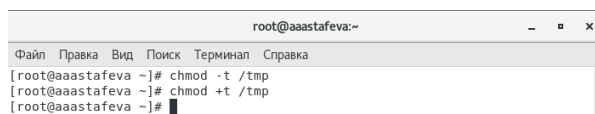
Figure 3.16: Снятие атрибута “t” с директории /tmp

15. От пользователя guest2 проверила, что атрибута t у директории /tmp нет: `ls -l / | grep tmp`. Повторила предыдущие шаги. Нам удалось удалить файл от имени пользователя, не являющегося его владельцем, также получилось выполнить дозапись в файл и замену текста в файле. (рис. 3.17).

```
guest2@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
[guest2@aaastafeva ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не позволена  
[guest2@aaastafeva ~]$ ls -l / | grep tmp  
drwxrwxrwx. 17 root root 4096 ноя  9 14:21 tmp  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test3  
[guest2@aaastafeva ~]$ echo "test3" >> /tmp/file01.txt  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test3  
test3  
[guest2@aaastafeva ~]$ echo "test" > /tmp/file01.txt  
[guest2@aaastafeva ~]$ cat /tmp/file01.txt  
test  
[guest2@aaastafeva ~]$ rm /tmp/file01.txt  
[guest2@aaastafeva ~]$ ls /tmp  
ssh-2lDChiaI62cL  
systemd-private-a99e6d8f613a439bb540a9b4a396c3b7-bolt.service-13rZKK  
systemd-private-a99e6d8f613a439bb540a9b4a396c3b7-colord.service-8PKJxU  
systemd-private-a99e6d8f613a439bb540a9b4a396c3b7-cups.service-AfnxYe  
systemd-private-a99e6d8f613a439bb540a9b4a396c3b7-fwupd.service-43b1sT  
systemd-private-a99e6d8f613a439bb540a9b4a396c3b7-rtkit-daemon.service-SFIdIW  
tracker-extract-files.1000  
tracker-extract-files.1001  
yum_save.tx.2021-11-09.13-13.jofRPK.yumtx  
[guest2@aaastafeva ~]$
```

Figure 3.17: Проверка

16. От суперпользователя вернула атрибут `t` на директорию `/tmp`: `chmod +t /tmp`.
(рис. 3.18).



```
root@aaastafeva:~  
Файл Правка Вид Поиск Терминал Справка  
[root@aaastafeva ~]# chmod -t /tmp  
[root@aaastafeva ~]# chmod +t /tmp  
[root@aaastafeva ~]#
```

Figure 3.18: Добавление атрибута “t” на директорию `/tmp`

4 Выводы

На основе проделанной работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. Получла практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.