

Отчёт по лабораторной работе №5

дисциплина: Информационная безопасность

Ерёменко Артём Геннадьевич, НПИбд-01-18

Содержание

Цель работы	3
Выполнение лабораторной работы	4
Выводы	9

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

1. Вошёл в систему от имени пользователя guest(рис. [-@fig:001]).

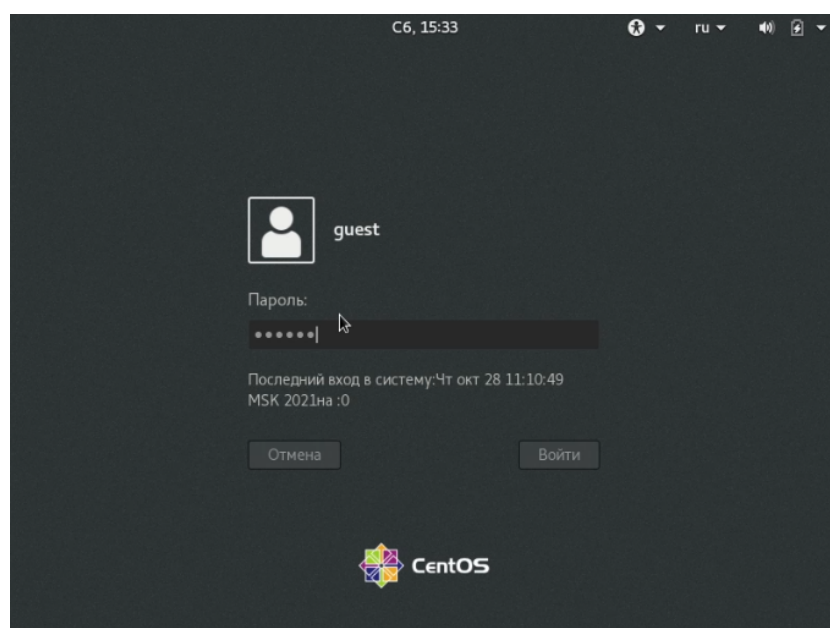
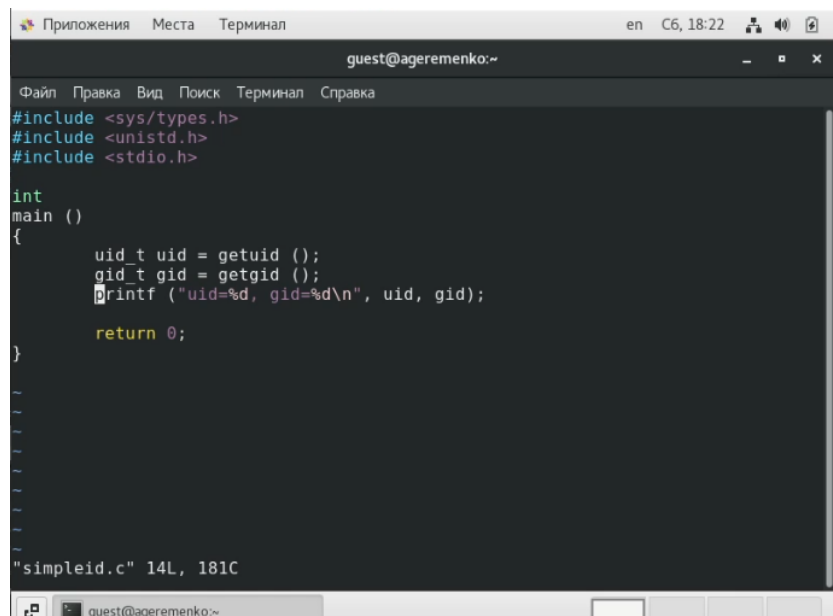


Рис. 0.1: Вход в систему под guest

2. Создал программу simpleid.c(рис. [-@fig:002]):

A screenshot of a terminal window titled "Приложения Места Терминал" with a sub-header "guest@ageremenko:~". The terminal shows the creation of a C file named "simpleid.c". The code includes headers for `<sys/types.h>`, `<unistd.h>`, and `<stdio.h>`. The `main` function calls `getuid()` and `getgid()` to retrieve the current user and group IDs, then prints them using `printf` and returns 0. The status bar at the bottom indicates the cursor is at line 14, column 181 of "simpleid.c".

```
guest@ageremenko:~  
Файл Правка Вид Поиск Терминал Справка  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = getuid ();  
    gid_t gid = getgid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
  
    return 0;  
}  
  
"simpleid.c" 14L, 181C
```

Рис. 0.2: Создание simpleid.c

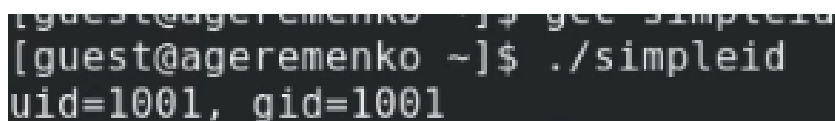
3. Скомпилировал программу и убедился, что файл программы создан([@fig:003]).

A screenshot of a terminal window showing the command to compile the program: `gcc simpleid.c -o simpleid`.

```
[guest@ageremenko ~]$ gcc simpleid.c -o simpleid
```

Рис. 0.3: Компиляция

4. Выполнил программу simpleid(рис. [-@fig:004]).

A screenshot of a terminal window showing the execution of the program. The command `./simpleid` is entered, and the output is `uid=1001, gid=1001`.

```
[guest@ageremenko ~]$ ./simpleid  
uid=1001, gid=1001
```

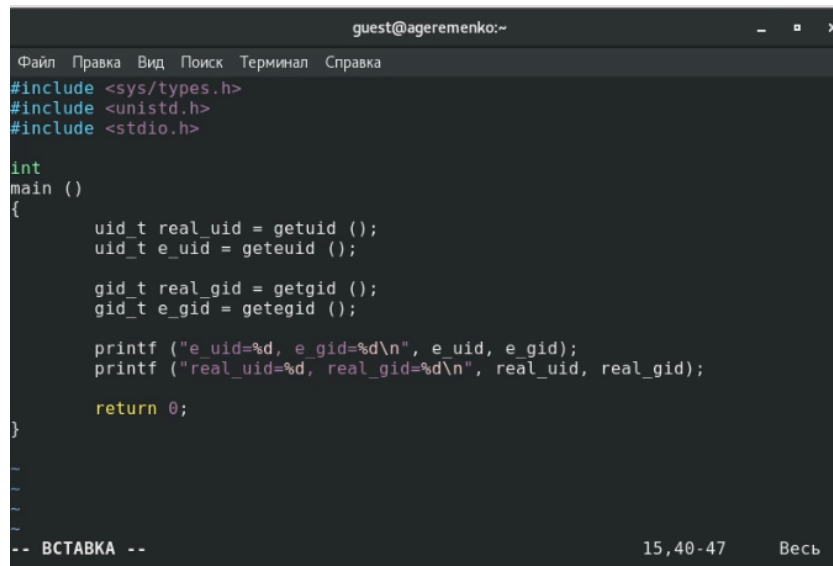
Рис. 0.4: Выполнение программы

5. Выполнил системную программу id(рис. [-@fig:005]).

```
[guest@ageremenko ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Рис. 0.5: Выполнение id

6. Усложнил программу, добавив вывод действительных идентификаторов(рис. [-@fig:006])



```
guest@ageremenko:~
Файл Правка Вид Поиск Терминал Справка
#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;
}

-- ВСТАВКА -- 15,40-47 Весь
```

Рис. 0.6: Усложнение программы

7. Скомпилировал и запустил simpleid2.c(рис. [-@fig:007])

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

Рис. 0.7: Компиляция и запуск

8. От имени суперпользователя выполнил команды(рис. [-@fig:008])

```

[root@ageremenko ~]# chown root:guest /home/guest/simpleid2
[root@ageremenko ~]# chmod u+s /home/guest/simpleid2

```

Рис. 0.8: Изменение прав

9. Использовал sudo (рис. [-@fig:009])

```

[ageremenko@ageremenko ~]$ sudo -i

```

Рис. 0.9: Использование sudo

10. Выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2. (рис. [-@fig:010])

```

[guest@ageremenko ~]$ ls -l simpleid2
-rwsrwxr-x. 1 root guest 7376 ноя 13 17:54 simpleid2

```

Рис. 0.10: Выполнение проверки

11. Запустил simpleid2 и id (рис. [-@fig:011])

```

[guest@ageremenko ~]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ageremenko ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023

```

Рис. 0.11: Запуск simpleid2 и id

12. Проделал тоже самое относительно SetGID-бита (рис. [-@fig:012])

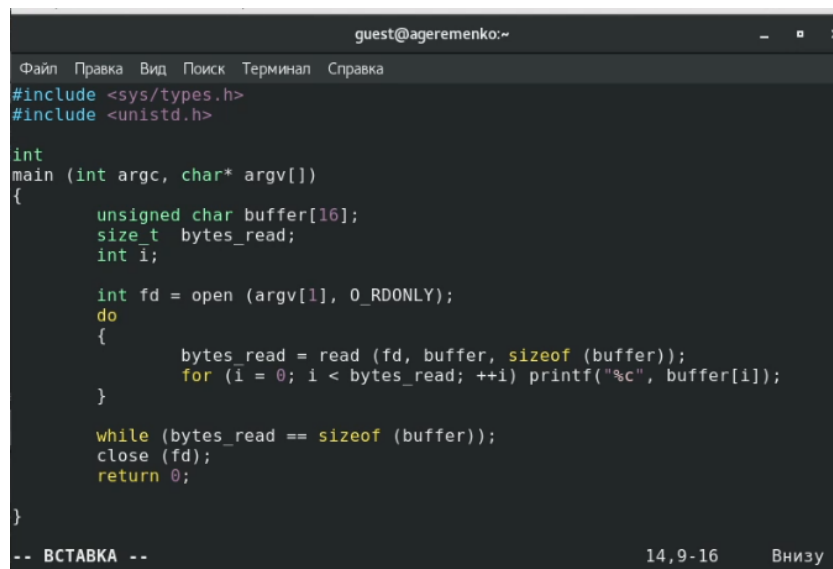
```

[root@ageremenko ~]# chmod g+s /home/guest/simpleid2

```

Рис. 0.12: Аналогичные действия для SetGID-бита

13. Создал программу readfile.c

A screenshot of a code editor window titled 'guest@ageremenko:~'. The editor has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The code is as follows:

```
#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;
}
```

At the bottom of the editor, there is a status bar with '-- ВСТАВКА --' on the left, '14,9-16' in the center, and 'Внизу' on the right.

Рис. 0.13: Создание readfile.c

14. Откомпилировал её

A screenshot of a terminal window showing the command to compile the program:

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

Рис. 0.14: Компиляция readfile.c

Выводы

Прodelав данную лабораторную работу я изученил механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Полученил практические навыки работы в кон- соли с дополнительными атрибутами. Рассмотрел работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.