
Front matter

lang: ru-RU

title: Лабораторная работа 5

subtitle: Основы информационной безопасности

author:

- Пинега Б.А.

institute:

- Российский университет дружбы народов, Москва, Россия

i18n babel

babel-lang: russian

babel-otherlangs: english

Formatting pdf

toc: false

toc-title: Содержание

slide_level: 2

aspectratio: 169

section-titles: true

theme: metropolis

header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- '\makeatletter'
- '\beamer@ignorenonframefalse'
- '\makeatother'

Докладчик

- * Пинега Белла Александровна

- * Студентка НБИБД-02-22

- * Российский университет дружбы народов

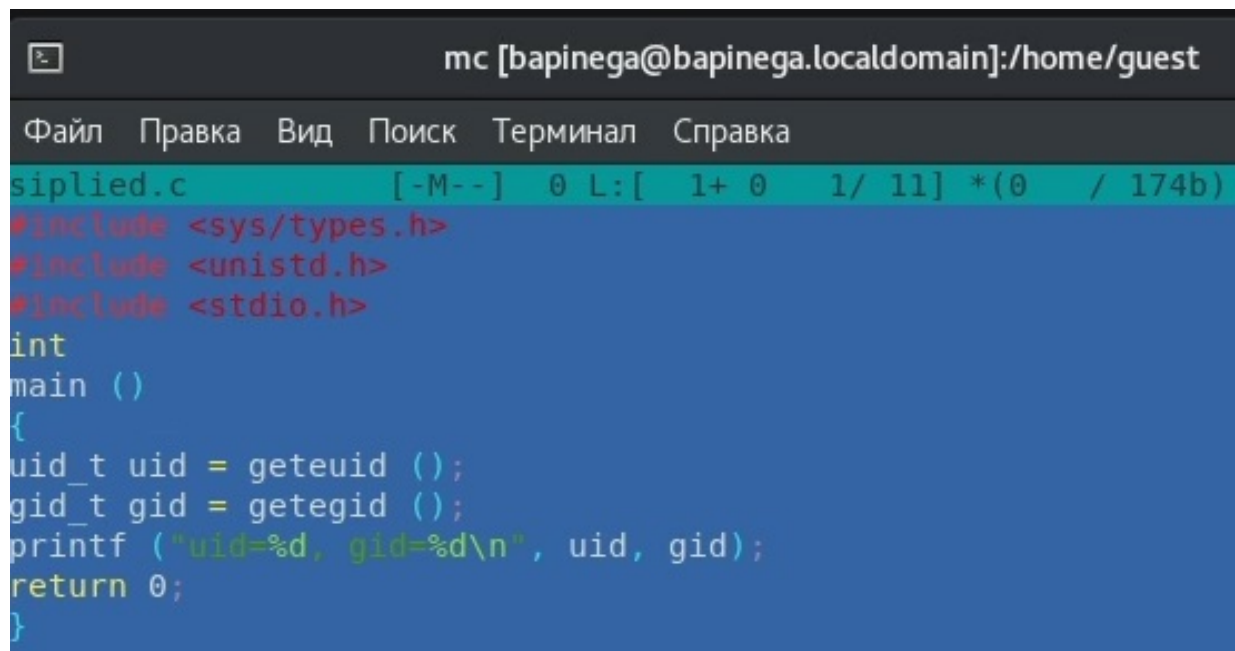
Цель работы

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

```
[bapinega@bapinega ~]$ cd /home/guest  
[bapinega@bapinega guest]$ touch siplied.c
```

{#fig:001 width=40%}



The screenshot shows a terminal window titled "mc [bapinega@bapinega.localdomain]:/home/guest". The window has a menu bar with "Файл", "Правка", "Вид", "Поиск", "Терминал", and "Справка". The file "siplied.c" is open, showing its metadata as "[-M- -] 0 L: [1+ 0 1/ 11] *(0 / 174b)". The code content is as follows:

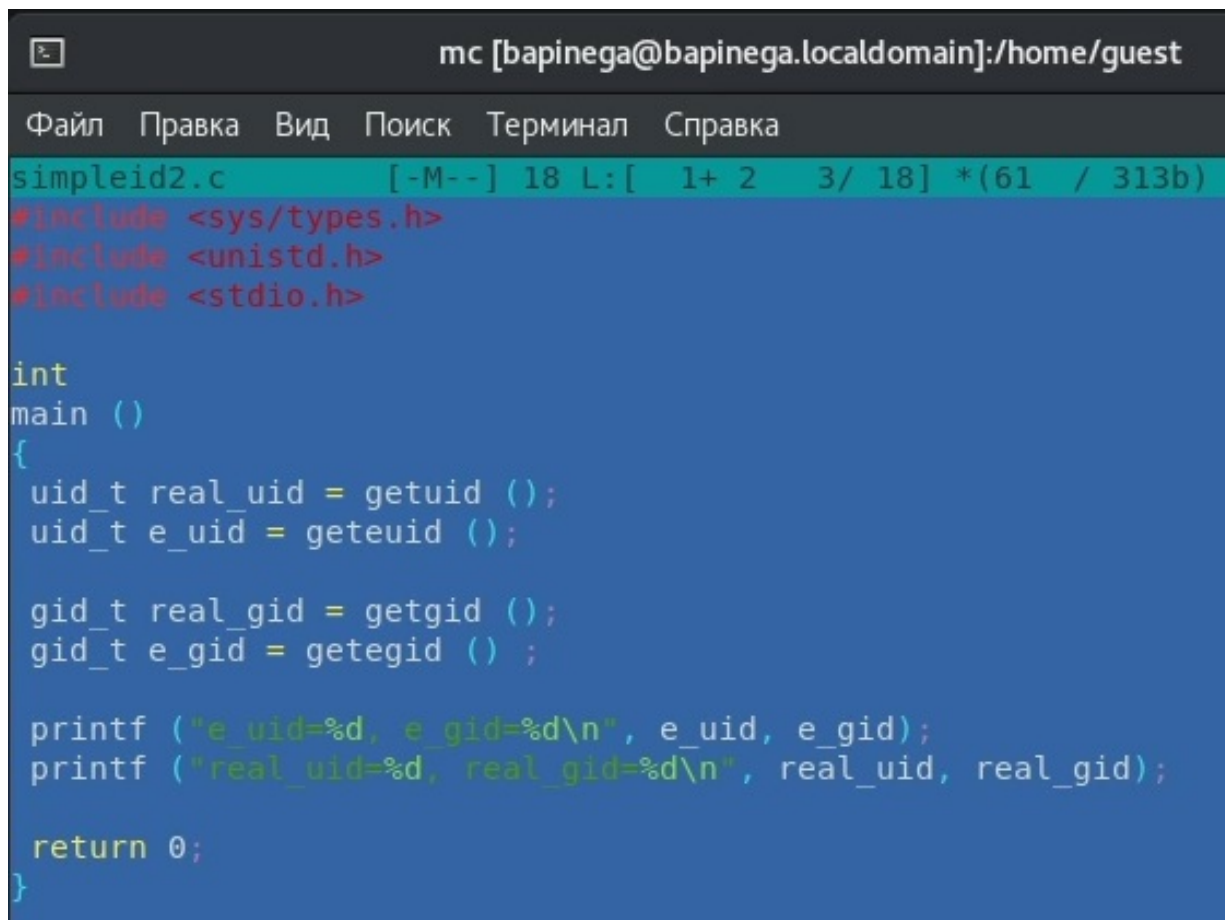
```
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
  
int  
main ()  
{  
    uid_t uid = geteuid ();  
    gid_t gid = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Скомпилирую программу

```
[bapinega@bapinega guest]$ mv siplied.c simpleid.c
[bapinega@bapinega guest]$ gcc simpleid.c -o simpleid
[bapinega@bapinega guest]$ ls
dir1  simpleid  simpleid.c
[bapinega@bapinega guest]$ ./sipleid
bash: ./sipleid: Нет такого файла или каталога
[bapinega@bapinega guest]$ ./simpleid
uid=1000, gid=1000
[bapinega@bapinega guest]$ id
uid=1000(bapinega) gid=1000(bapinega) группы=1000(bapinega),10(wheel) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

{#fig:003 width=40%}

Усложню программу



```
mc [bapinega@bapinega.localdomain]:/home/guest
Файл  Правка  Вид  Поиск  Терминал  Справка
simpleid2.c  [-M--] 18 L:[ 1+ 2 3/ 18] *(61 / 313b)
#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;
}
```

Скомпилирую и запущу

```
[bapinega@bapinega guest]$ gcc simpleid2.c -o simpleid2  
[bapinega@bapinega guest]$ ./simpleid2  
e_uid=1000, e_gid=1000  
real_uid=1000, real_gid=1000
```

{#fig:005 width=70%}

От имени суперпользователя выполню команды

```
[bapinega@bapinega guest]$ su  
Пароль:  
[root@bapinega guest]# chown root:guest /home/guest/simpleid2  
[root@bapinega guest]# chmod u+s /home/guest/simpleid2  
[root@bapinega guest]#
```

{#fig:006 width=70%}

Сменю владельца у файла и изменю права

```
[bapinega@bapinega guest]$ su
Пароль:
[root@bapinega guest]# chown root:root readfile
[root@bapinega guest]# chmod -rwx readfile.c
chmod: readfile.c: новые права доступа ----w----, а не -----
[root@bapinega guest]# chmod u+s readfile
[root@bapinega guest]# exit
exit
[bapinega@bapinega guest]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[bapinega@bapinega guest]$ ./readfile readfile.c
bash: ./readfile: Отказано в доступе
[bapinega@bapinega guest]$ su
Пароль:
[root@bapinega guest]# chown root:guest /home/guest/readfile/c
chown: невозможно получить доступ к '/home/guest/readfile/c': Это не каталог
[root@bapinega guest]# chown root:guest /home/guest/readfile.c
[root@bapinega guest]# chmod 700 readfile.c
```

```
[root@bapinega guest]# ./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;
```

Программа readfile может прочитать файл /etc/shadow

```
[root@bapinega guest]# ./readfile /etc/shadow
root:$6$PAbFRHdtXhkq6Ra$5x9HS1lwR/XHg20t9wWBH7J3AFP7ugdascT60oIxnMbLCC
n0qJc20Z0N2n3o7P8g6GE2/4GL27030:0:99999:7:::
bin:!:19326:0:99999:7:::
daemon:!:19326:0:99999:7:::
adm:!:19326:0:99999:7:::
lp:!:19326:0:99999:7:::
sync:!:19326:0:99999:7:::
shutdown:!:19326:0:99999:7:::
halt:!:19326:0:99999:7:::
mail:!:19326:0:99999:7:::
operator:!:19326:0:99999:7:::
games:!:19326:0:99999:7:::
ftp:!:19326:0:99999:7:::
nobody:!:19326:0:99999:7:::
dbus:!!:19768:::
systemd-coredump:!!:19768:::
systemd-resolve:!!:19768:::
tss:!!:19768:::
polkitd:!!:19768:::
geoclue:!!:19768:::
unbound:!!:19768:::
rtkit:!!:19768:::
pipewire:!!:19768:::
dnsmasq:!!:19768:::
clevis:!!:19768:::
usbmuxd:!!:19768:::
gluster:!!:19768:::
rpc:!!:19768:0:99999:7:::
chrony:!!:19768:::
avahi:!!:19768:::
ssslauth:!!:19768:::
libstoragemgmt:!!:19768:::
sssd:!!:19768:::
qemu:!!:19768:::
cockpit-ws:!!:19768:::
cockpit-wsinstance:!!:19768:::
colord:!!:19768:::
rpcuser:!!:19768:::
pulse:!!:19768:::
setroubleshoot:!!:19768:::
flatpak:!!:19768:::
gdm:!!:19768:::
gnome-initial-setup:!!:19768:::
design:!!:19768:::
sshd:!!:19768:::
tcpdump:!!:19768:::
```

Выводы

Я изучила механизмы 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 влияния бита Sticky на запись и удаление файлов