

Лабораторная работа №5

Основы информационной безопасности

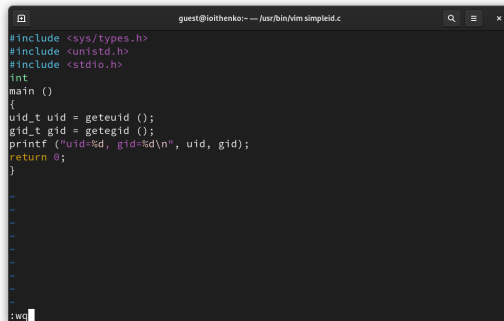
Ищенко Ирина

Российский университет дружбы народов, Москва, Россия

- Ищенко Ирина Олеговна
- НПИбд-02-22

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.

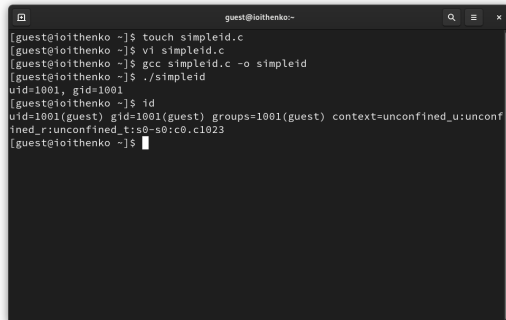
Выполнение лабораторной работы



The image shows a terminal window with a dark background. The title bar at the top reads "guest@ioithenko:~ — /usr/bin/vim simpleid.c". The code is displayed in a light-colored font with syntax highlighting. The code includes headers for `<sys/types.h>`, `<unistd.h>`, and `<stdio.h>`. It defines an `int` type and a `main ()` function. Inside the function, it declares `uid_t uid` and `gid_t gid`, then calls `geteuid ()` and `getegid ()` to retrieve the effective user and group IDs. These values are printed to the standard output using `printf ("uid=%d, gid=%d\n", uid, gid);`. The function ends with `return 0;`. The cursor is at the end of the last line of code, and the command `:wq` is visible in the bottom left corner of the editor.

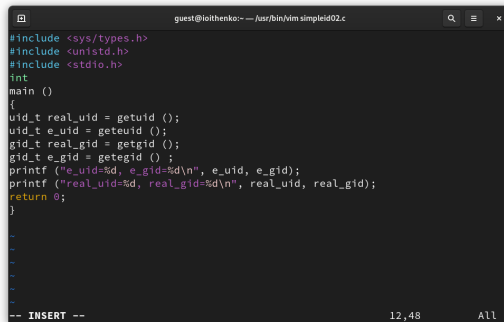
```
guest@ioithenko:~ — /usr/bin/vim simpleid.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int
main ()
{
    uid_t uid = geteuid ();
    gid_t gid = getegid ();
    printf ("uid=%d, gid=%d\n", uid, gid);
    return 0;
}
:wq
```

Рис. 1: simpleid.c

A terminal window titled 'guest@ioithenko:~' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[guest@ioithenko ~]$ touch simpleid.c
[guest@ioithenko ~]$ vi simpleid.c
[guest@ioithenko ~]$ gcc simpleid.c -o simpleid
[guest@ioithenko ~]$ ./simpleid
uid=1001, gid=1001
[guest@ioithenko ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@ioithenko ~]$
```

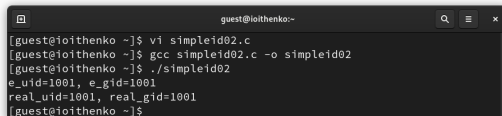
Рис. 2: Выполнение simpleid



```
guest@ioithenko:~ -- /usr/bin/vim simpleid02.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;
}

-- INSERT --
```

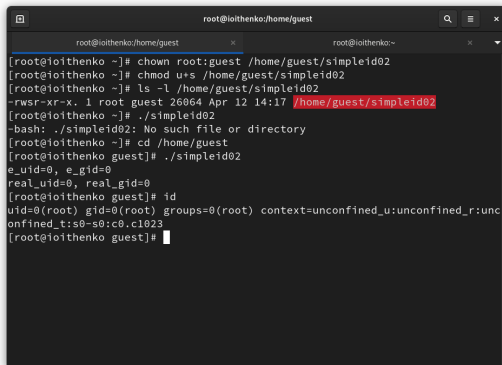
Рис. 3: simpleid02



```
guest@ioithenko:~  
[guest@ioithenko ~]$ vi simpleid02.c  
[guest@ioithenko ~]$ gcc simpleid02.c -o simpleid02  
[guest@ioithenko ~]$ ./simpleid02  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@ioithenko ~]$
```

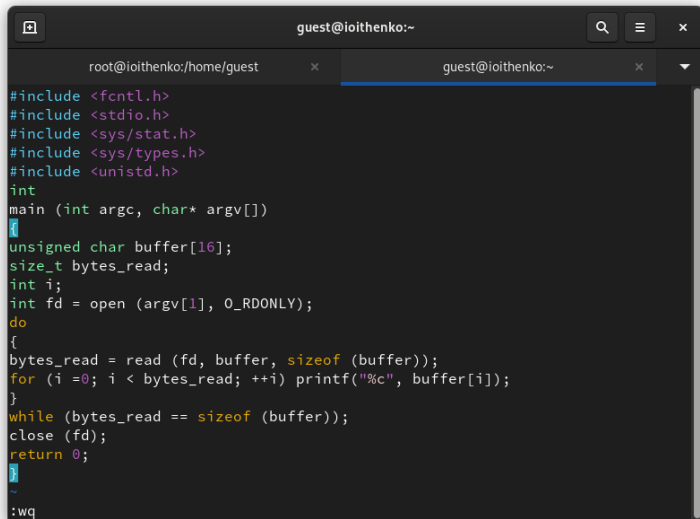
Рис. 4: Выполнение simpleid02

Смена владельца и атрибут



```
root@ioithenko/home/guest
root@ioithenko/home/guest
[root@ioithenko ~]# chown root:guest /home/guest/simpleid02
[root@ioithenko ~]# chmod u+s /home/guest/simpleid02
[root@ioithenko ~]# ls -l /home/guest/simpleid02
-rwsr-xr-x. 1 root guest 26064 Apr 12 14:17 /home/guest/simpleid02
[root@ioithenko ~]# ./simpleid02
-bash: ./simpleid02: No such file or directory
[root@ioithenko ~]# cd /home/guest
[root@ioithenko guest]# ./simpleid02
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@ioithenko guest]# id
uid=0(root) gid=0(root) groups=0(root) context=unconfined_u:unconfined_r:unc
onfined_t:s0-s0:c0.c1023
[root@ioithenko guest]#
```

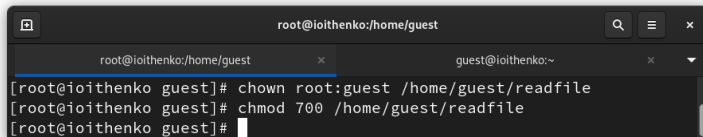
Рис. 5: Смена владельца и атрибут



A terminal window titled "guest@ioithenko:~" with a search icon, a menu icon, and a close icon in the top right corner. The window has two tabs: "root@ioithenko:/home/guest" and "guest@ioithenko:~". The active tab shows the following C code:

```
#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;
}
~
:wq
```

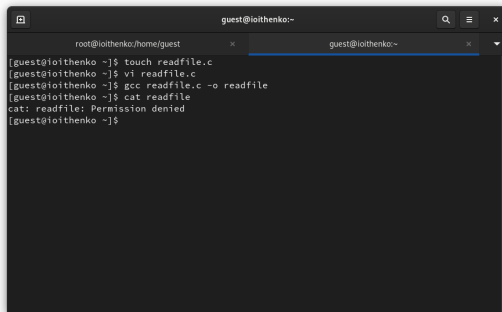
Изменение владельца и прав



```
root@ioithenko:/home/guest
[root@ioithenko guest]# chown root:guest /home/guest/readfile
[root@ioithenko guest]# chmod 700 /home/guest/readfile
[root@ioithenko guest]#
```

The image shows a terminal window with a dark background. The title bar at the top reads 'root@ioithenko:/home/guest'. Below the title bar, there are two tabs: 'root@ioithenko:/home/guest' (active) and 'guest@ioithenko:~'. The terminal content shows three lines of commands and their prompts: '[root@ioithenko guest]# chown root:guest /home/guest/readfile', '[root@ioithenko guest]# chmod 700 /home/guest/readfile', and '[root@ioithenko guest]#' followed by a cursor. The window includes standard UI elements like a search icon, a menu icon, and a close button in the top right corner.

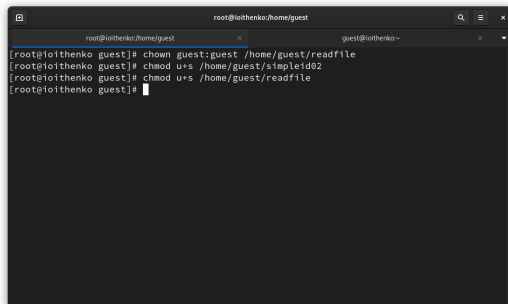
Рис. 7: Изменение владельца и прав



A terminal window titled 'guest@ioithenko:~' with two tabs: 'root@ioithenko:/home/guest' and 'guest@ioithenko:~'. The active tab shows the following commands and output:

```
[guest@ioithenko ~]$ touch readfile.c
[guest@ioithenko ~]$ vi readfile.c
[guest@ioithenko ~]$ gcc readfile.c -o readfile
[guest@ioithenko ~]$ cat readfile
cat: readfile: Permission denied
[guest@ioithenko ~]$
```

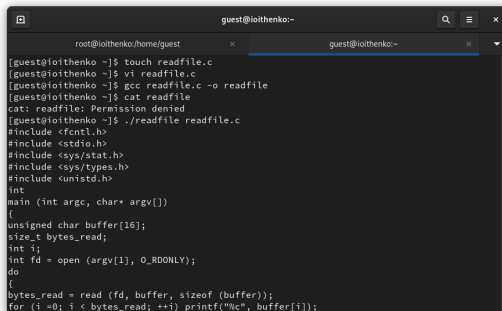
Рис. 8: Отказ в чтении



A terminal window titled 'root@ioithenko:/home/guest' with two tabs. The active tab is 'root@ioithenko:/home/guest'. The other tab is 'guest@ioithenko ~'. The terminal shows the following commands and output:

```
[root@ioithenko guest]# chown guest:guest /home/guest/readfile
[root@ioithenko guest]# chmod u+s /home/guest/simpleid02
[root@ioithenko guest]# chmod u+s /home/guest/readfile
[root@ioithenko guest]#
```

Рис. 9: Смена владельца и атрибута



```
guest@ioithenko:~  
root@ioithenko:/home/guest  
[guest@ioithenko ~]$ touch readfile.c  
[guest@ioithenko ~]$ vi readfile.c  
[guest@ioithenko ~]$ gcc readfile.c -o readfile  
[guest@ioithenko ~]$ cat readfile  
cat: readfile: Permission denied  
[guest@ioithenko ~]$ ./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]);
```

Рис. 10: Выполнение проверки

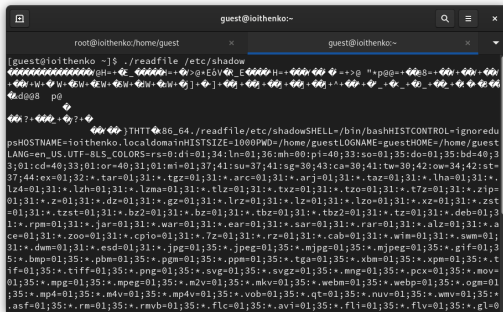
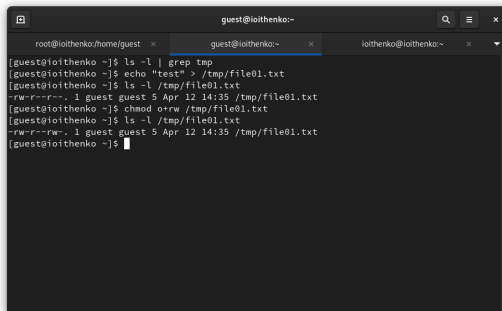
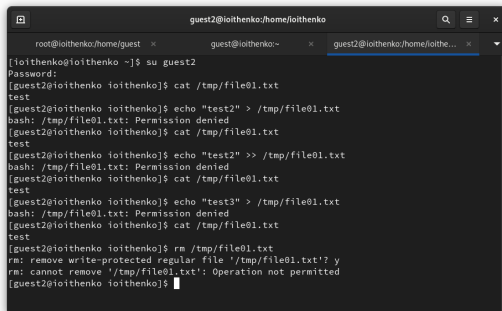


Рис. 11: Выполнение проверки



```
guest@ioithenko:~  
root@ioithenko/home/guest x guest@ioithenko:~ x ioithenko@ioithenko:~ x  
[guest@ioithenko ~]$ ls -l | grep tmp  
[guest@ioithenko ~]$ echo "test" > /tmp/file01.txt  
[guest@ioithenko ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Apr 12 14:35 /tmp/file01.txt  
[guest@ioithenko ~]$ chmod o+rw /tmp/file01.txt  
[guest@ioithenko ~]$ ls -l /tmp/file01.txt  
-rw-r--rw-. 1 guest guest 5 Apr 12 14:35 /tmp/file01.txt  
[guest@ioithenko ~]$
```

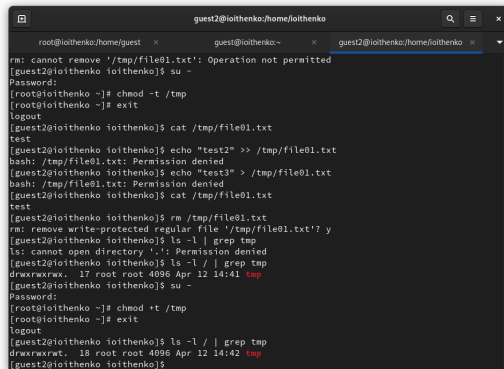
Рис. 12: Права на файл

A terminal window titled 'guest2@ioithenko:/home/ioithenko' with three tabs. The active tab shows a sequence of commands and outputs. The user 'ioithenko' switches to 'guest2' and attempts to write to '/tmp/file01.txt'. The first attempt fails with 'Permission denied'. The second attempt, using 'echo "test2" >> /tmp/file01.txt', also fails with 'Permission denied'. The third attempt, using 'echo "test3" > /tmp/file01.txt', also fails with 'Permission denied'. Finally, the user attempts to remove the file with 'rm /tmp/file01.txt', which fails with 'rm: cannot remove '/tmp/file01.txt': Operation not permitted'.

```
guest2@ioithenko:/home/ioithenko
root@ioithenko:/home/guest x guest@ioithenko:~ x guest2@ioithenko:/home/ioithe... x
[ioithenko@ioithenko ~]$ su guest2
Password:
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@ioithenko ioithenko]$
```

Рис. 13: Проверка атрибута

Проверка снятия атрибута



```
guest2@ioithenko:/home/ioithenko
root@ioithenko:/home/guest x guest@ioithenko:~ x guest2@ioithenko:/home/ioithenko x
rm: cannot remove '/tmp/file01.txt': Operation not permitted
[guest2@ioithenko ioithenko]$ su -
Password:
[root@ioithenko ~]# chmod -t /tmp
[root@ioithenko ~]# exit
logout
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ echo "test2" >> /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ echo "test3" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@ioithenko ioithenko]$ cat /tmp/file01.txt
test
[guest2@ioithenko ioithenko]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@ioithenko ioithenko]$ ls -l | grep tmp
ls: cannot open directory '.': Permission denied
[guest2@ioithenko ioithenko]$ ls -l / | grep tmp
drwxrwxrwx. 17 root root 4096 Apr 12 14:41 tmp
[guest2@ioithenko ioithenko]$ su -
Password:
[root@ioithenko ~]# chmod +t /tmp
[root@ioithenko ~]# exit
logout
[guest2@ioithenko ioithenko]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Apr 12 14:42 tmp
[guest2@ioithenko ioithenko]$
```

Рис. 14: Проверка снятия атрибута

В ходе лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. 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.