

# Лабораторная работа №5

## Информационная безопасность

---

Александрова УВ

12.03.2024

Российский университет дружбы народов, Москва, Россия

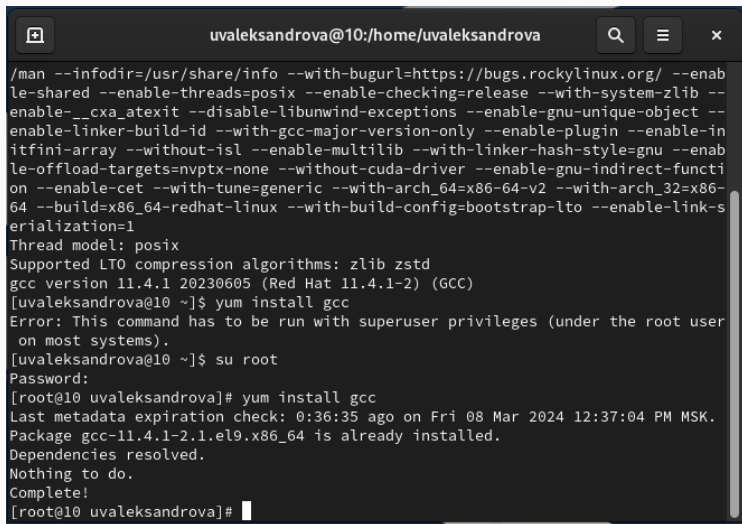
## Цель работы

---

Целью работы является изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов.

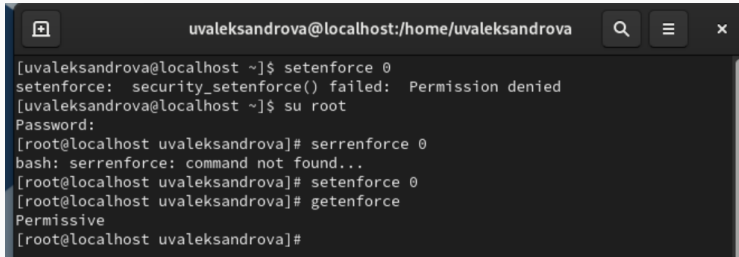
## Подготовка к выполнению работы

---



```
uvaleksandrova@10:/home/uvaleksandrova
/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
Thread model: posix
Supported LTO compression algorithms: zlib zstd
gcc version 11.4.1 20230605 (Red Hat 11.4.1-2) (GCC)
[uvaleksandrova@10 ~]$ yum install gcc
Error: This command has to be run with superuser privileges (under the root user on most systems).
[uvaleksandrova@10 ~]$ su root
Password:
[root@10 uvaleksandrova]# yum install gcc
Last metadata expiration check: 0:36:35 ago on Fri 08 Mar 2024 12:37:04 PM MSK.
Package gcc-11.4.1-2.1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@10 uvaleksandrova]#
```

Figure 1: Проверка установки ПО



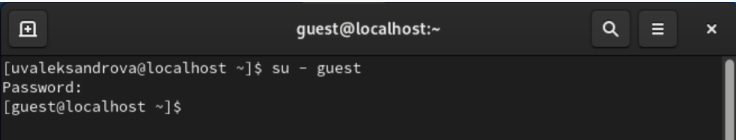
```
uvaleksandrova@localhost:/home/uvaleksandrova

[uvaleksandrova@localhost ~]$ setenforce 0
setenforce: security_setenforce() failed: Permission denied
[uvaleksandrova@localhost ~]$ su root
Password:
[root@localhost uvaleksandrova]# serrenforce 0
bash: serrenforce: command not found...
[root@localhost uvaleksandrova]# setenforce 0
[root@localhost uvaleksandrova]# getenforce
Permissive
[root@localhost uvaleksandrova]#
```

Figure 2: setenforce 0

## Выполнение лабораторной работы

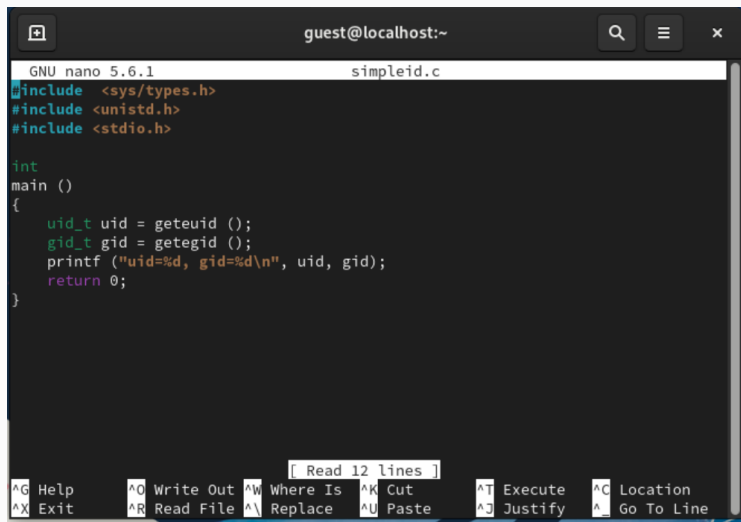
---

A terminal window with a dark background. The title bar shows a window icon, the text 'guest@localhost:~', and search, menu, and close buttons. The terminal content shows a user switch command and its execution.

```
guest@localhost:~  
[uvaleksandrova@localhost ~]$ su - guest  
Password:  
[guest@localhost ~]$
```

Figure 3: Вход в систему от другого пользователя





```
guest@localhost:~  
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;  
}
```

[ Read 12 lines ]

<b>^G</b> Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut	<b>^T</b> Execute	<b>^C</b> Location
<b>^X</b> Exit	<b>^R</b> Read File	<b>^I</b> Replace	<b>^U</b> Paste	<b>^J</b> Justify	<b>^_</b> Go To Line

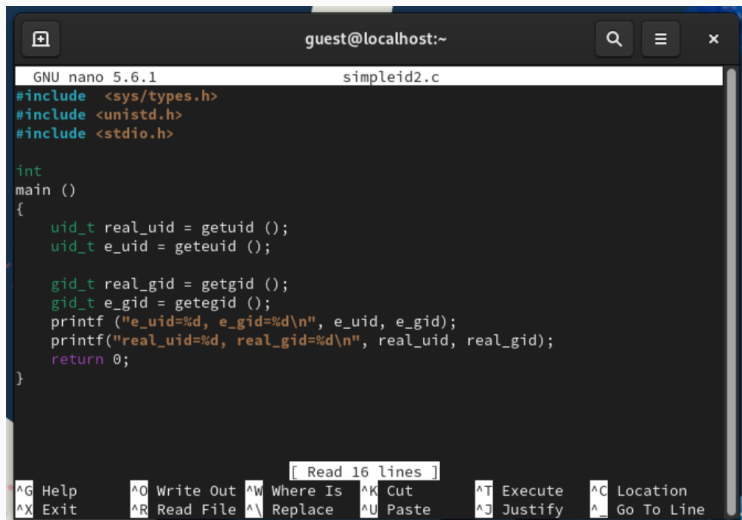
Figure 4: Заполнение элементарной программы

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid  
[guest@localhost ~]$ ./simpleid  
uid=1001, gid=1001
```

Figure 5: Компиляция и запуск программы

```
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Figure 6: Команда id



```
GNU nano 5.6.1 simpleid2.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

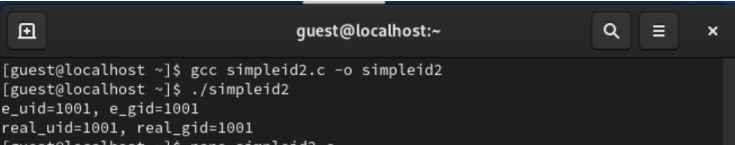
int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();

    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;
}
```

[ Read 16 lines ]

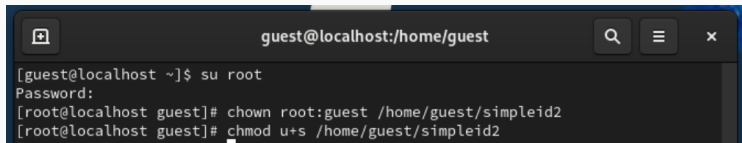
<b>^G</b> Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut	<b>^T</b> Execute	<b>^C</b> Location
<b>^X</b> Exit	<b>^R</b> Read File	<b>^I</b> Replace	<b>^U</b> Paste	<b>^J</b> Justify	<b>^_</b> Go To Line

Figure 7: Заполнение программы



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

Figure 8: Компиляция и запуск программы



A terminal window titled 'guest@localhost:/home/guest' with search, menu, and close buttons. The terminal shows a user switching to root and then changing permissions for a file.

```
[guest@localhost ~]$ su root
Password:
[root@localhost guest]# chown root:guest /home/guest/simpleid2
[root@localhost guest]# chmod u+s /home/guest/simpleid2
```

Figure 9: Поменяла владельца программы

```
[guest@localhost ~]$ ls -l simpleid2  
-rwsr-xr-x. 1 root guest 26064 Mar 11 16:20 simpleid2  
[guest@localhost ~]$
```

Figure 10: ls -l

```
[guest@localhost ~]$ id  
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest@localhost ~]$
```

Figure 11: Сравнение результатов

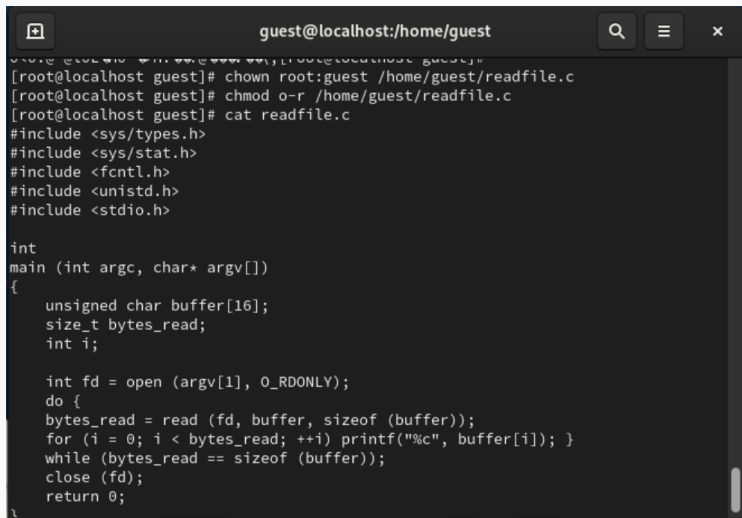


```
[guest@localhost ~]$ cat readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.h>

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;
}
```

Figure 12: Создание программы



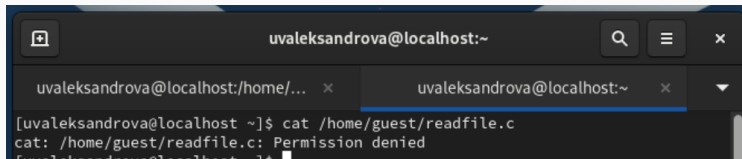
A terminal window titled 'guest@localhost:/home/guest' with search, menu, and close buttons. It shows the execution of commands to set permissions on 'readfile.c' and the content of the file, which is a C program for reading a file.

```
guest@localhost:/home/guest
[root@localhost guest]# chown root:guest /home/guest/readfile.c
[root@localhost guest]# chmod o-r /home/guest/readfile.c
[root@localhost guest]# cat readfile.c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdio.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;
}
```

Figure 13: Компиляция программы



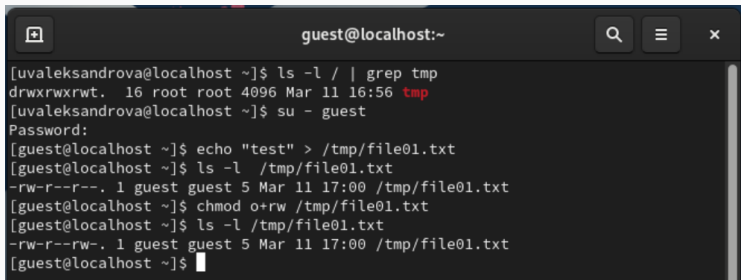
A terminal window titled 'uvaleksandrova@localhost:~' with search, menu, and close buttons. It shows a command prompt where the user has entered 'cat /home/guest/readfile.c'. The output is 'cat: /home/guest/readfile.c: Permission denied'. The terminal has a dark theme and a tab bar at the top.

```
uvaleksandrova@localhost:~  
[uvaleksandrova@localhost ~]$ cat /home/guest/readfile.c  
cat: /home/guest/readfile.c: Permission denied
```

Figure 14: Смена владельца

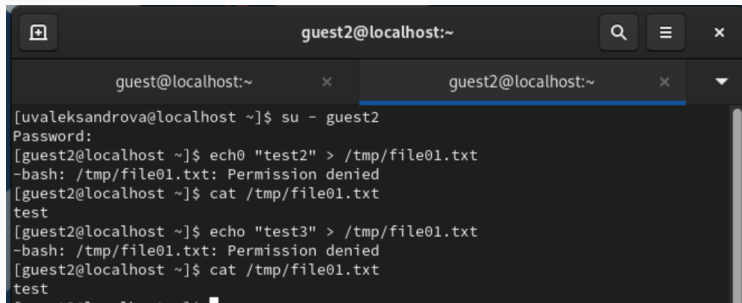
[illegible]

Figure 15: Результат работы программы



```
guest@localhost:~  
[uvaleksandrova@localhost ~]$ ls -l / | grep tmp  
drwxrwxrwt. 16 root root 4096 Mar 11 16:56 tmp  
[uvaleksandrova@localhost ~]$ su - guest  
Password:  
[guest@localhost ~]$ echo "test" > /tmp/file01.txt  
[guest@localhost ~]$ ls -l /tmp/file01.txt  
-rw-r--r--. 1 guest guest 5 Mar 11 17:00 /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 Mar 11 17:00 /tmp/file01.txt  
[guest@localhost ~]$
```

Figure 16: Выполнение задач

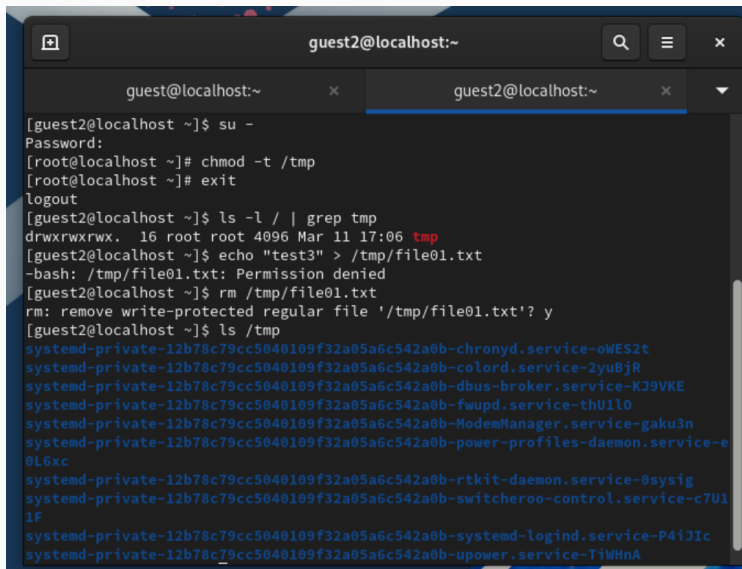


A terminal window titled 'guest2@localhost:~' showing a sequence of commands and their outputs. The user switches to the 'guest2' user and attempts to create and read a file in the /tmp directory. The file is created successfully, but subsequent attempts to write to it are denied due to the sticky bit. The output of the cat command shows the file contains 'test'.

```
guest@localhost:~  
[uvaleksandrova@localhost ~]$ su - guest2  
Password:  
[guest2@localhost ~]$ echo "test2" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@localhost ~]$ cat /tmp/file01.txt  
test  
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@localhost ~]$ cat /tmp/file01.txt  
test
```

Figure 17: Работа с файлами

## Исследование Sticky-бита



```
guest2@localhost:~  
[guest2@localhost ~]$ su -  
Password:  
[root@localhost ~]# chmod -t /tmp  
[root@localhost ~]# exit  
logout  
[guest2@localhost ~]$ ls -l / | grep tmp  
drwxrwxrwx. 16 root root 4096 Mar 11 17:06 tmp  
[guest2@localhost ~]$ echo "test3" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@localhost ~]$ rm /tmp/file01.txt  
rm: remove write-protected regular file '/tmp/file01.txt'? y  
[guest2@localhost ~]$ ls /tmp  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-chronyd.service-oWES2t  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-colord.service-2yuBjR  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-dbus-broker.service-KJ9VKE  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-fwupd.service-thU1l0  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-ModemManager.service-gaku3n  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-power-profiles-daemon.service-e  
0L6xc  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-rtkit-daemon.service-0sysig  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-switcheroo-control.service-c7U1  
1F  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-systemd-logind.service-P4iJIc  
systemd-private-12b78c79cc5040109f32a05a6c542a0b-upower.service-TiWHnA
```

```
[root@localhost ~]# chmod +t /tmp  
[root@localhost ~]# exit
```

Figure 19: Возвращение Sticky



## Выводы

---

Я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практических навыков работы в консоли с дополнительными атрибутами.