

Отчет к лабораторной работе №5

Common information

discipline: Основы информационной безопасности

group: НПМбд-02-21

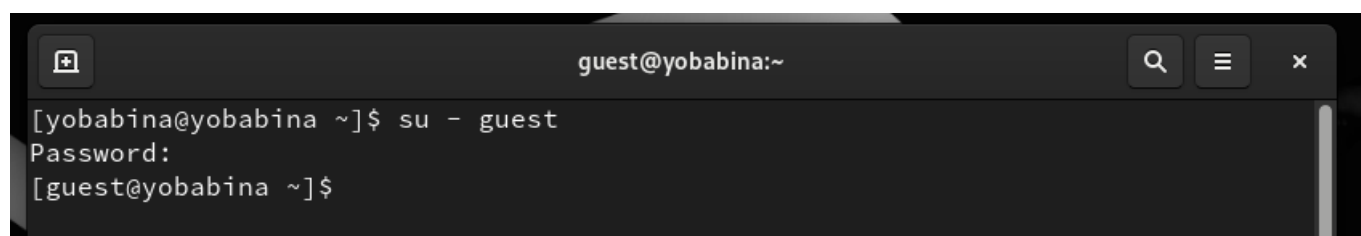
author: Бабина Ю. О.

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 работы

Войдем в систему от имени пользователя guest.



```
guest@yobabina:~  
[yobabina@yobabina ~]$ su - guest  
Password:  
[guest@yobabina ~]$
```

Создадим программу simpleid.c:



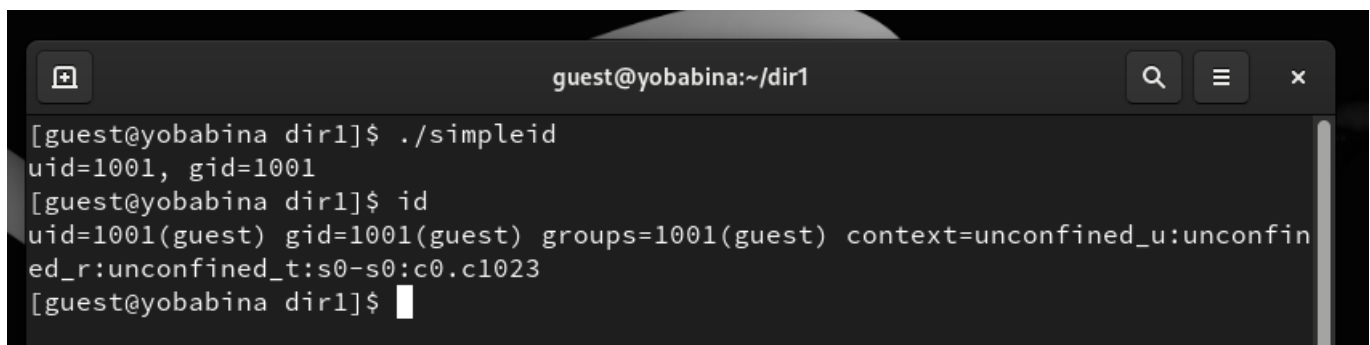
```
guest@yobabina:~/dir1  
GNU nano 5.6.1 simpleid.c  
#include <sys/types.h>  
#include <unistd.h>  
#include <stdio.h>  
int main () {  
    uid_t uid = geteuid ();  
    gid_t gid = getegid ();  
    printf ("uid=%d, gid=%d\n", uid, gid);  
    return 0;  
}
```

Скомпилируем программу и убедимся, что файл программы создан:



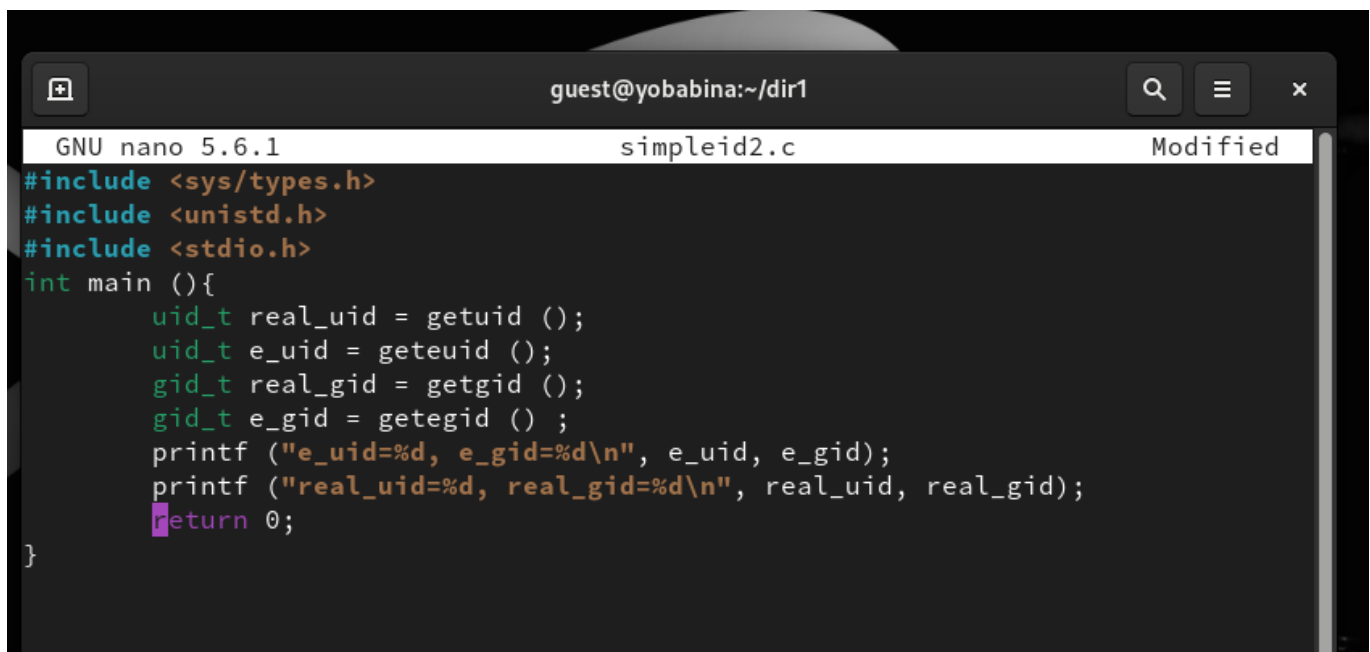
```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ gcc simpleid.c -o simpleid
[guest@yobabina dir1]$ ls
file1  simpleid  simpleid.c
[guest@yobabina dir1]$
```

Выполним программу simpleid и системную программу id:



```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ./simpleid
uid=1001, gid=1001
[guest@yobabina dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@yobabina dir1]$
```

Усложним программу, добавив вывод действительных идентификаторов. Далее получившуюся программу назовем simpleid2.c.



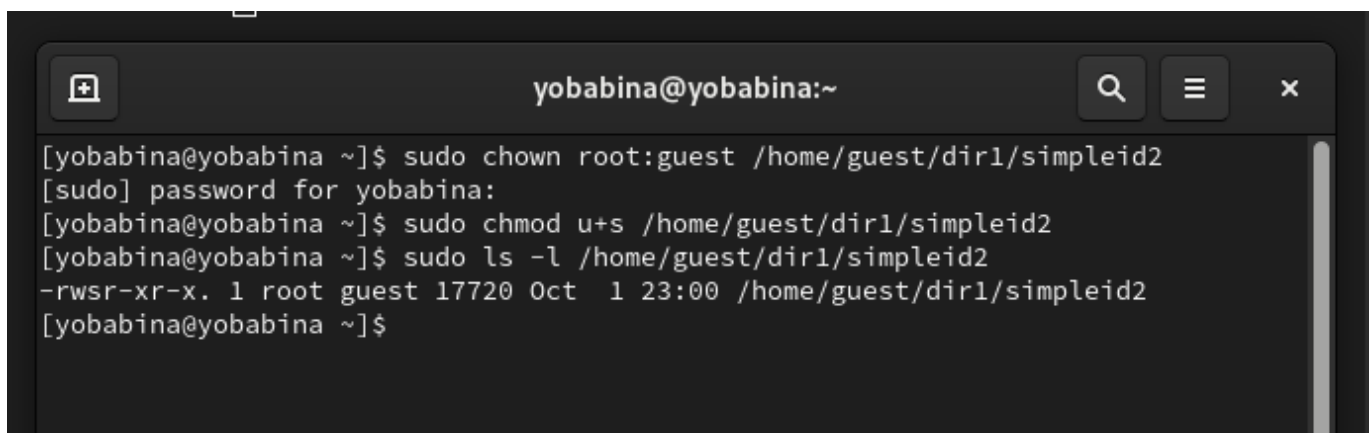
```
GNU nano 5.6.1      simpleid2.c      Modified
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
int main (){
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();
    printf ("e_uid=%d, e_gid=%d\n", e_uid, e_gid);
    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
    return 0;
}
```

Скомпилируем и запустим simpleid2.c:



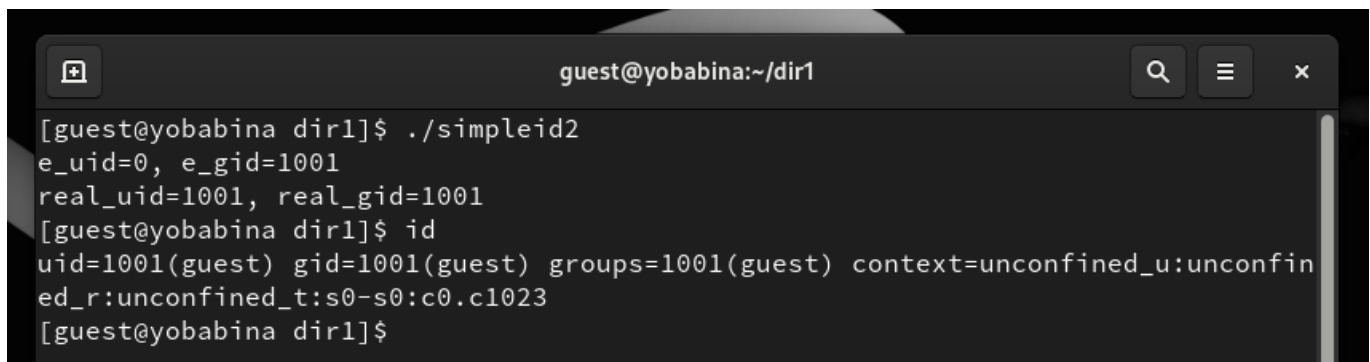
```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ gcc simpleid2.c -o simpleid2
[guest@yobabina dir1]$ ls
file1  simpleid  simpleid2  simpleid2.c  simpleid.c
[guest@yobabina dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yobabina dir1]$
```

От имени суперпользователя выполним команды: "chown root:guest /home/guest/simpleid2" и "chmod u+s /home/guest/simpleid2". Выполним проверку правильности установки новых атрибутов и смены владельца файла simpleid2:



```
yobabina@yobabina:~
[yobabina@yobabina ~]$ sudo chown root:guest /home/guest/dir1/simpleid2
[sudo] password for yobabina:
[yobabina@yobabina ~]$ sudo chmod u+s /home/guest/dir1/simpleid2
[yobabina@yobabina ~]$ sudo ls -l /home/guest/dir1/simpleid2
-rwsr-xr-x. 1 root guest 17720 Oct  1 23:00 /home/guest/dir1/simpleid2
[yobabina@yobabina ~]$
```

Запустим simpleid2 и id:



```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yobabina dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@yobabina dir1]$
```

Прделаем тоже самое относительно SetGID-бита:



The image shows two terminal windows. The top window, titled 'guest@yobabina:~/dir1', shows the execution of './simpleid2' which outputs 'e_uid=1001, e_gid=1001' and 'real_uid=1001, real_gid=1001'. It then shows the output of 'id', which is 'uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023'. The bottom window, titled 'yobabina@yobabina:~', shows the execution of 'sudo chown root:guest /home/guest/dir1/simpleid2' and 'sudo chmod g+s /home/guest/dir1/simpleid2'.

```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@yobabina dir1]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[guest@yobabina dir1]$

yobabina@yobabina:~
[yobabina@yobabina ~]$ sudo chown root:guest /home/guest/dir1/simpleid2
[yobabina@yobabina ~]$ sudo chmod g+s /home/guest/dir1/simpleid2
[yobabina@yobabina ~]$
```

Создадим программу readfile.c:



The image shows a terminal window titled 'guest@yobabina:~/dir1' with the GNU nano 5.6.1 editor open to a file named 'readfile.c'. The code is as follows:

```
GNU nano 5.6.1 readfile.c Modified
#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;
}
```

At the bottom of the terminal, there is a row of keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^_ Replace, ^U Paste, ^J Justify, ^_ Go To Line.

Откомпилируем её с помощью команды: "gcc readfile.c -o readfile":



The image shows a terminal window titled 'guest@yobabina:~/dir1' with the following commands and output:

```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ gcc readfile.c -o readfile
[guest@yobabina dir1]$ ls
file1 readfile readfile.c simpleid simpleid2 simpleid2.c simpleid.c
[guest@yobabina dir1]$
```

Сменим владельца у файла readfile.c (или любого другого текстового файла в системе) и изменим права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог:

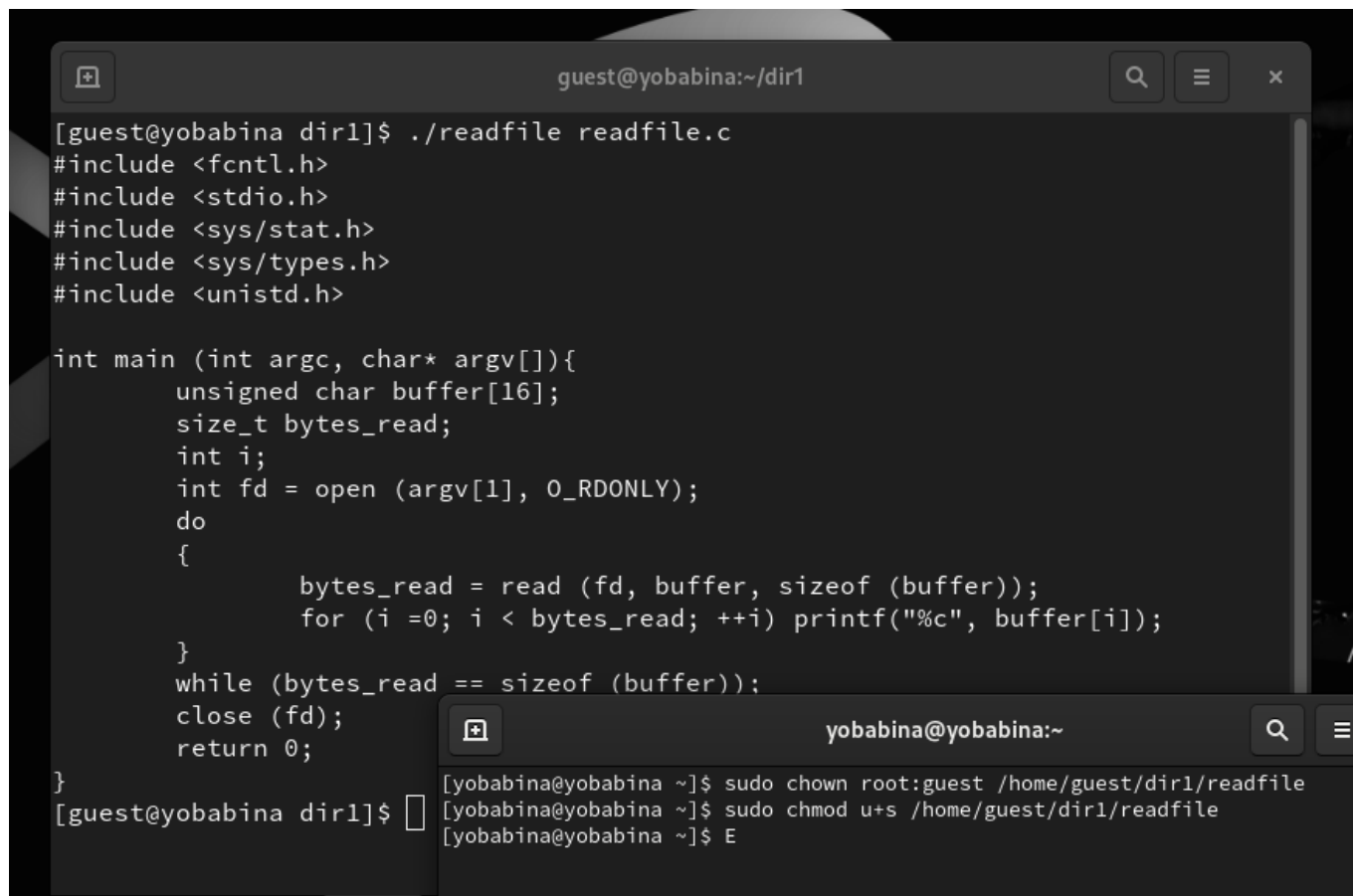
```
yobabina@yobabina:~$ sudo chown root:guest /home/guest/dir1/readfile.c
[yobabina@yobabina ~]$ sudo chmod 700 /home/guest/dir1/readfile.c
[yobabina@yobabina ~]$
```

Проверим, что пользователь guest не может прочитать файл readfile.c. Далее сменим у программы readfile владельца и установим SetUID-бит.

```
guest@yobabina:~/dir1$ ./readfile readfile.c
9%V@x /|x>@9%)
R]T 9% @ P9%p@h
p@
a86_64./readfilereadfile.cSHELL=/bin/bashHISTCONTROL=ign
oredupsHISTSIZ=1000HOSTNAME=yobabina.localdomainPWD=/home/guest/dir1LOGNAME=gue
stXAUTHORITY=/home/guest/.xauth3XudYlHOME=/home/guestLANG=en_GB.UTF-8LS_COLORS=r
s=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:01:cd=40;33:01:o
r=40;31:01:mi=01;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=
01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.
lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;
31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;3
1:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=
01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.s
ar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31
:*.rz=01;31:*.cab=01;31:*.wim=01;31:*.swm=01;31:*.dwm=01;31:*.esd=01;31:*.jpg=01
;35:*.jpeg=01;35:*.mjpg=01;35:*.mjpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:
*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=0
1;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.
mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.webp=01;35:*.ogm=0
1;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.w
```

```
== sizeof (buffer));
yobabina@yobabina:~$ sudo chown root:guest /home/guest/dir1/readfile
[yobabina@yobabina ~]$ sudo chmod u+s /home/guest/dir1/readfile
[yobabina@yobabina ~]$ E
```

Проверим, может ли программа readfile прочитать файл readfile.c:

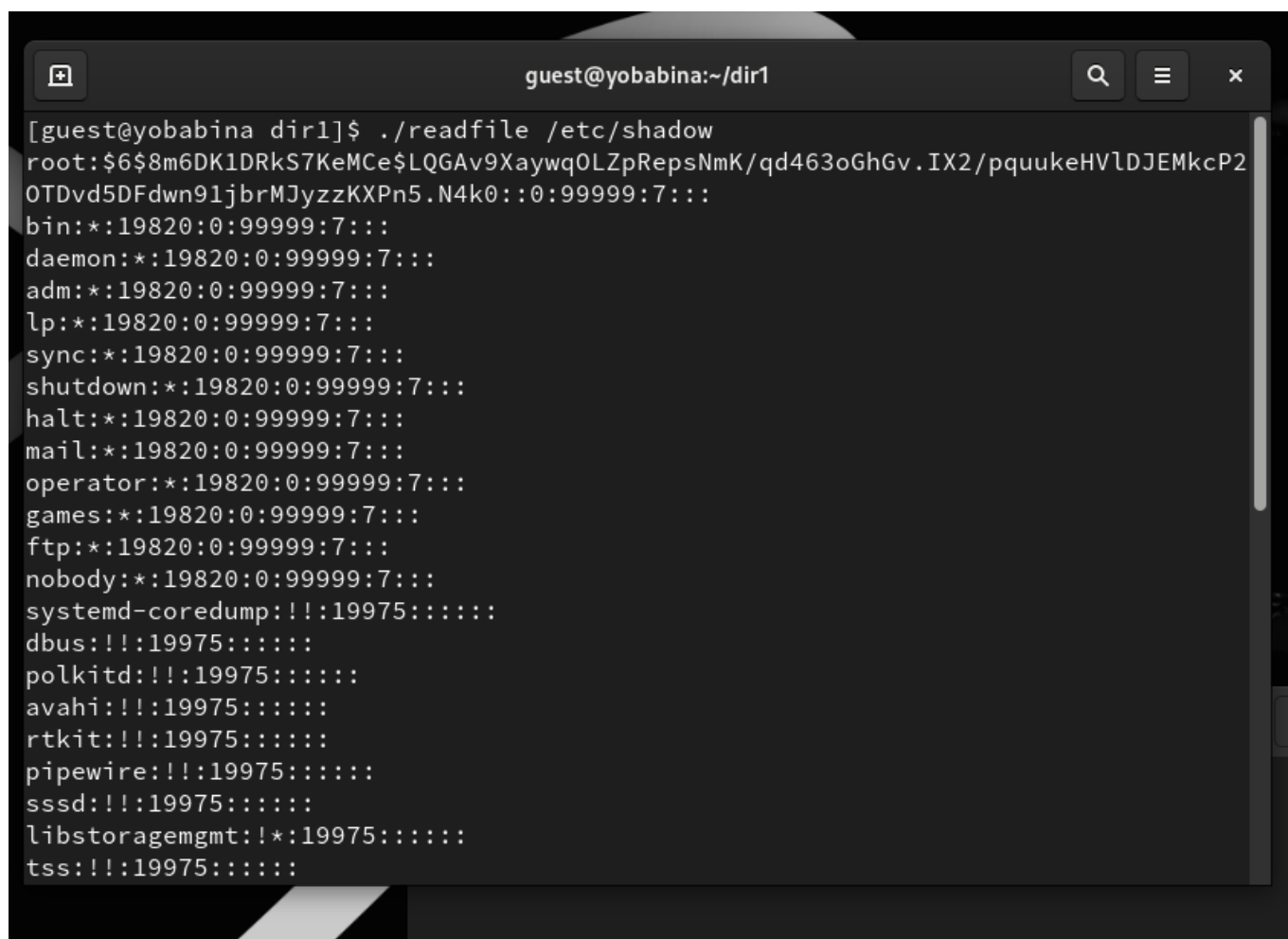


```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

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;
}
[guest@yobabina dir1]$

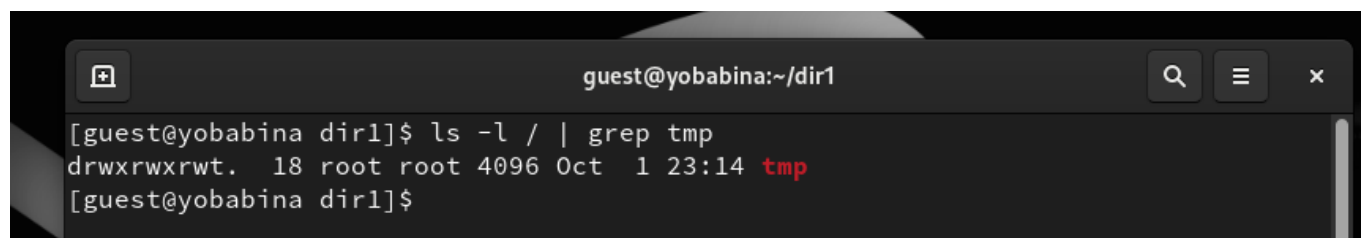
yobabina@yobabina:~
[yobabina@yobabina ~]$ sudo chown root:guest /home/guest/dir1/readfile
[yobabina@yobabina ~]$ sudo chmod u+s /home/guest/dir1/readfile
[yobabina@yobabina ~]$ E
```

Проверим, может ли программа readfile прочитать файл /etc/shadow:



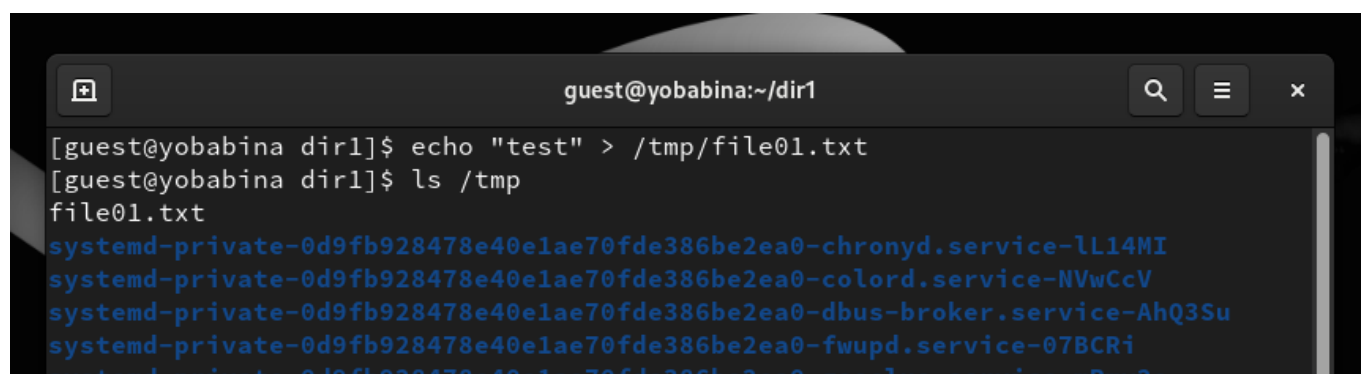
```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ./readfile /etc/shadow
root:$6$8m6DK1DRkS7KeMCe$LQGA9Xaywq0LZpRepsNmK/qd463oGhGv.IX2/pquukeHVlDJEMkcP2
OTDvd5DFdwn91jbrMJyzzKXPn5.N4k0::0:99999:7:::
bin:*:19820:0:99999:7:::
daemon:*:19820:0:99999:7:::
adm:*:19820:0:99999:7:::
lp:*:19820:0:99999:7:::
sync:*:19820:0:99999:7:::
shutdown:*:19820:0:99999:7:::
halt:*:19820:0:99999:7:::
mail:*:19820:0:99999:7:::
operator:*:19820:0:99999:7:::
games:*:19820:0:99999:7:::
ftp:*:19820:0:99999:7:::
nobody:*:19820:0:99999:7:::
systemd-coredump:!!:19975:::::::
dbus:!!:19975:::::::
polkitd:!!:19975:::::::
avahi:!!:19975:::::::
rtkit:!!:19975:::::::
pipewire:!!:19975:::::::
sssd:!!:19975:::::::
libstoragemgmt:!*:19975:::::::
tss:!!:19975:::::::
```

Выясним, установлен ли атрибут Sticky на директории /tmp, для этого выполним команду: "ls -l / | grep tmp".



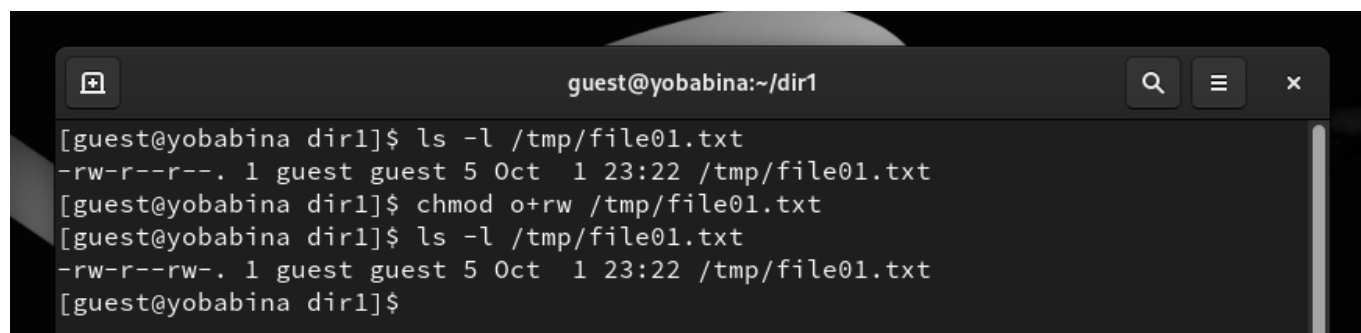
```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ls -l / | grep tmp
drwxrwxrwt. 18 root root 4096 Oct  1 23:14 tmp
[guest@yobabina dir1]$
```

От имени пользователя guest создадим файл file01.txt в директории /tmp со словом test с помощью команды: "echo "test" > /tmp/file01.txt":



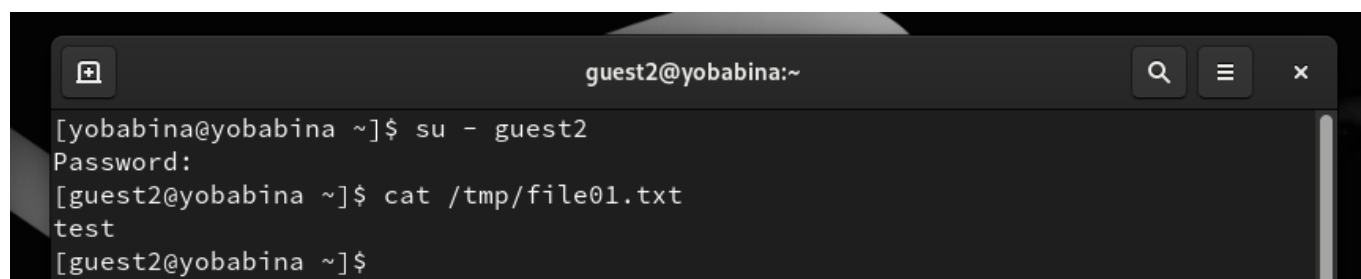
```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ echo "test" > /tmp/file01.txt
[guest@yobabina dir1]$ ls /tmp
file01.txt
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-chrond.service-LL14MI
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-colord.service-NVwCcV
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-dbus-broker.service-AhQ3Su
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-fwupd.service-07BCri
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-gssd.service-mBmy2g
```

Просмотрим атрибуты у только что созданного файла и разрешим чтение и запись для категории пользователей «все остальные»:



```
guest@yobabina:~/dir1
[guest@yobabina dir1]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Oct  1 23:22 /tmp/file01.txt
[guest@yobabina dir1]$ chmod o+rw /tmp/file01.txt
[guest@yobabina dir1]$ ls -l /tmp/file01.txt
-rw-r--rw-. 1 guest guest 5 Oct  1 23:22 /tmp/file01.txt
[guest@yobabina dir1]$
```

От пользователя guest2 (не являющегося владельцем) попробуем прочитать файл /tmp/file01.txt с помощью команды: "cat /tmp/file01.txt":



```
guest2@yobabina:~
[yobabina@yobabina ~]$ su - guest2
Password:
[guest2@yobabina ~]$ cat /tmp/file01.txt
test
[guest2@yobabina ~]$
```

Далее от пользователя guest2 попробуем дозаписать в файл /tmp/file01.txt слово test2:

```
guest2@yobabina:~  
[guest2@yobabina ~]$ echo "test2" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@yobabina ~]$ D
```

Проверим содержимое файла командой: "cat /tmp/file01.txt":

```
guest2@yobabina:~  
[guest2@yobabina ~]$ cat /tmp/file01.txt  
test  
[guest2@yobabina ~]$
```

Также попробуем записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию:

```
guest2@yobabina:~  
[guest2@yobabina ~]$ echo "test3" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@yobabina ~]$ cat /tmp/file01.txt  
test  
[guest2@yobabina ~]$
```

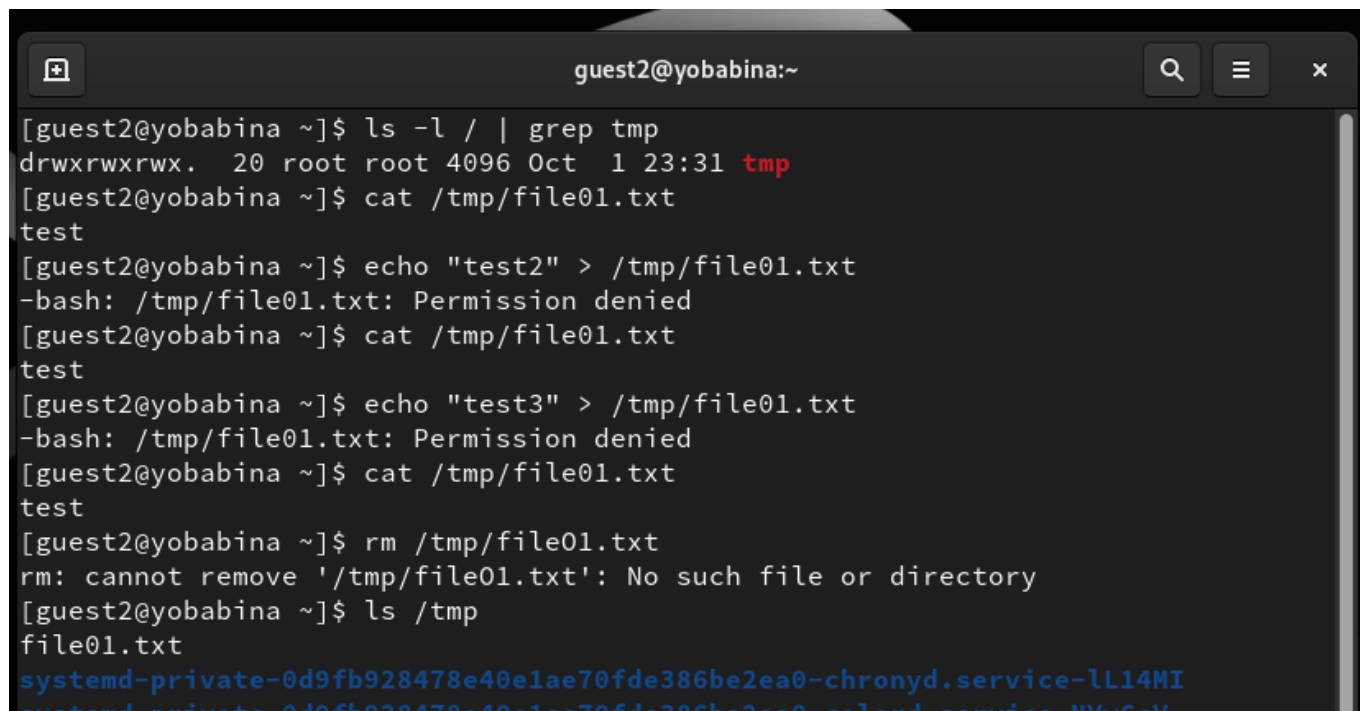
От пользователя guest2 попробуем удалить файл /tmp/file01.txt:

```
guest2@yobabina:~  
[guest2@yobabina ~]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': No such file or directory  
[guest2@yobabina ~]$ ls /tmp  
file01.txt  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-chrond.service-LL14MI  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-colord.service-NVwCcV  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-dbus-broker.service-AhQ3Su  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-fwupd.service-07BCRi
```

Повысим свои права до суперпользователя и выполним после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp. Далее покинем режим суперпользователя:

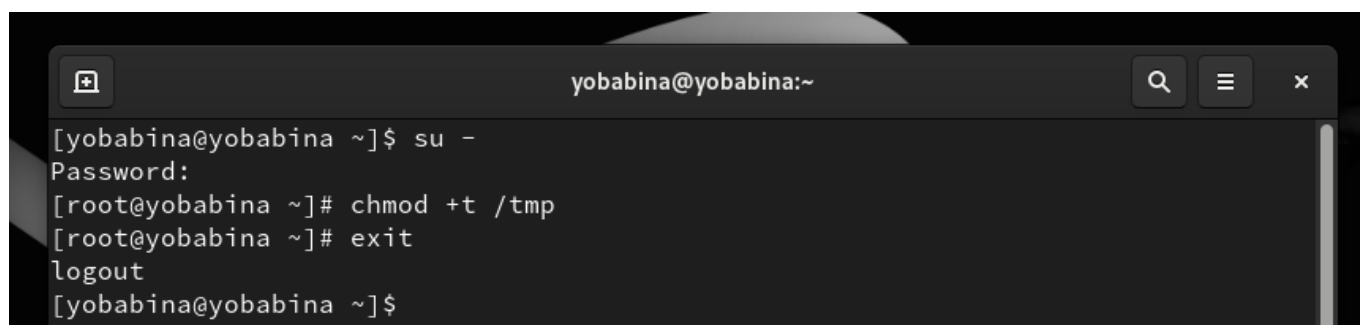
```
yobabina@yobabina:~  
[yobabina@yobabina ~]$ su -  
Password:  
[root@yobabina ~]# chmod -t /tmp  
[root@yobabina ~]# exit  
logout  
[yobabina@yobabina ~]$
```


От пользователя guest2 проверим, что атрибута t у директории /tmp нет. Далее повторим предыдущие шаги:



```
guest2@yobabina:~  
[guest2@yobabina ~]$ ls -l / | grep tmp  
drwxrwxrwx. 20 root root 4096 Oct 1 23:31 tmp  
[guest2@yobabina ~]$ cat /tmp/file01.txt  
test  
[guest2@yobabina ~]$ echo "test2" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@yobabina ~]$ cat /tmp/file01.txt  
test  
[guest2@yobabina ~]$ echo "test3" > /tmp/file01.txt  
-bash: /tmp/file01.txt: Permission denied  
[guest2@yobabina ~]$ cat /tmp/file01.txt  
test  
[guest2@yobabina ~]$ rm /tmp/file01.txt  
rm: cannot remove '/tmp/file01.txt': No such file or directory  
[guest2@yobabina ~]$ ls /tmp  
file01.txt  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-chrond.service-LL14MI  
systemd-private-0d9fb928478e40e1ae70fde386be2ea0-colord.service-MVwCcV
```

Повысим свои права до суперпользователя и вернем атрибут t на директорию /tmp:



```
yobabina@yobabina:~  
[yobabina@yobabina ~]$ su -  
Password:  
[root@yobabina ~]# chmod +t /tmp  
[root@yobabina ~]# exit  
logout  
[yobabina@yobabina ~]$
```

Вывод

В рамках выполнения данной 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.

Список литературы

- <https://habr.com/ru/articles/469667/>
- <https://www.golinuxcloud.com/sticky-bit-linux/>
- <https://rockylinux.org/>