Защита лабораторной работы №5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Бармина Ольга Константиновна

2022 Sep 21th

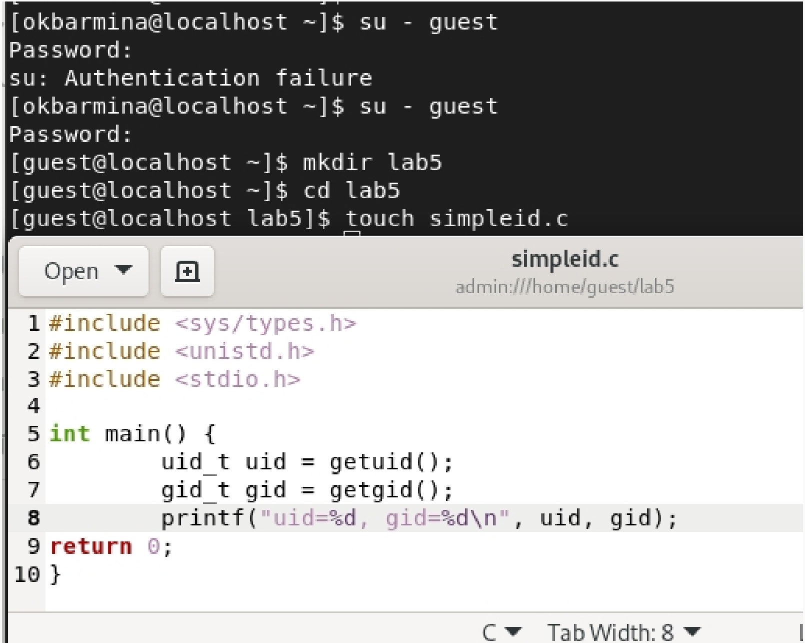
RUDN University, Moscow, Russian Federation

Результат выполнения лабораторной работы №5

Цель выполнения лабораторной 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.

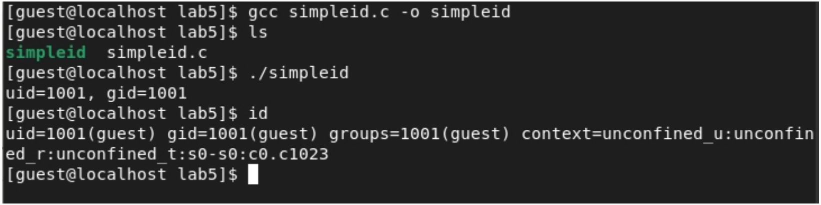
Результат выполнения лабораторной работы



```
[okbarmina@localhost ~]$ su - guest
Password:
su: Authentication failure
[okbarmina@localhost ~]$ su - guest
Password:
[guest@localhost ~]$ mkdir lab5
[guest@localhost ~]$ cd lab5
[guest@localhost lab5]$ touch simpleid.c
```

```c
#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;
}
```

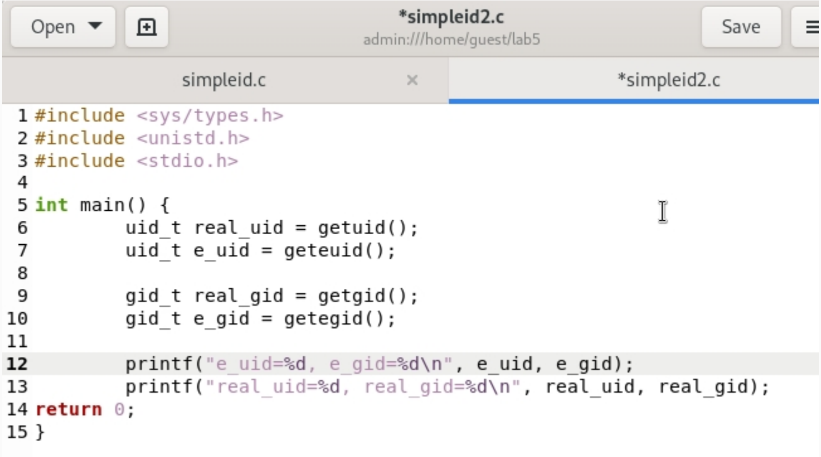
Результат выполнения лабораторной работы



```
[guest@localhost lab5]$ gcc simpleid.c -o simpleid
[guest@localhost lab5]$ ls
simpleid  simpleid.c
[guest@localhost lab5]$ ./simpleid
uid=1001, gid=1001
[guest@localhost lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost lab5]$ 
```

Figure 2: рис 2. Запуск simpleid.c

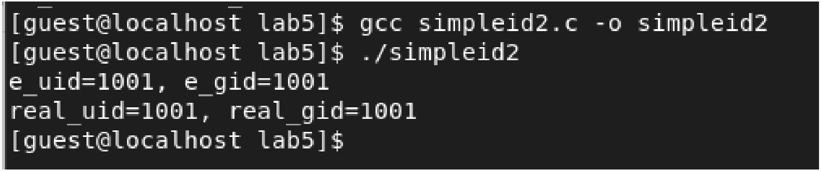
Результат выполнения лабораторной работы



```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;
}
```

Figure 3: рис 3. Дополнение программы

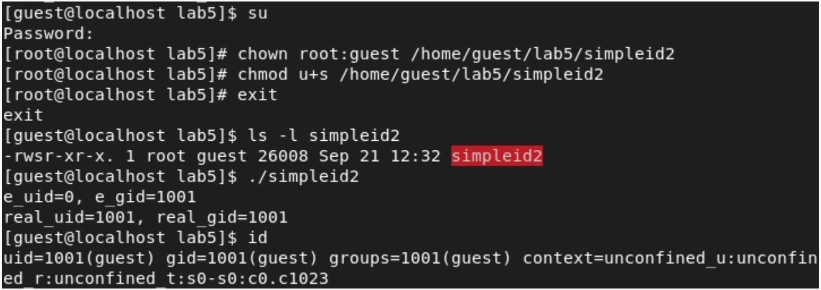
Результат выполнения лабораторной работы



```
[guest@localhost lab5]$ gcc simpleid2.c -o simpleid2
[guest@localhost lab5]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost lab5]$
```

Figure 4: рис 4. Запуск simpleid2.c

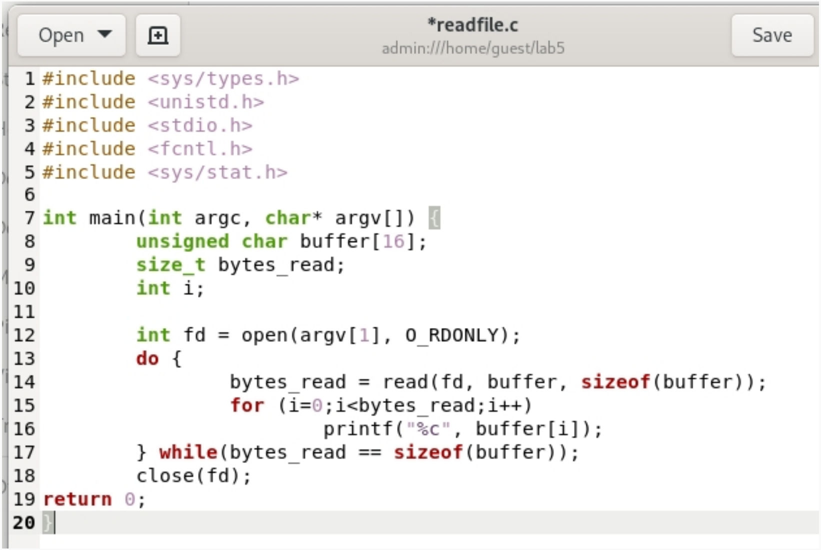
Результат выполнения лабораторной работы



```
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest /home/guest/lab5/simpleid2
[root@localhost lab5]# chmod u+s /home/guest/lab5/simpleid2
[root@localhost lab5]# exit
exit
[guest@localhost lab5]$ ls -l simpleid2
-rwsr-xr-x. 1 root guest 26008 Sep 21 12:32 simpleid2
[guest@localhost lab5]$ ./simpleid2
e_uid=0, e_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost lab5]$ id
uid=1001(guest) gid=1001(guest) groups=1001(guest) context=unconfined_u:unconfin
ed_r:unconfined_t:s0-s0:c0.c1023
```

Figure 5: рис 5. Команды суперпользователя

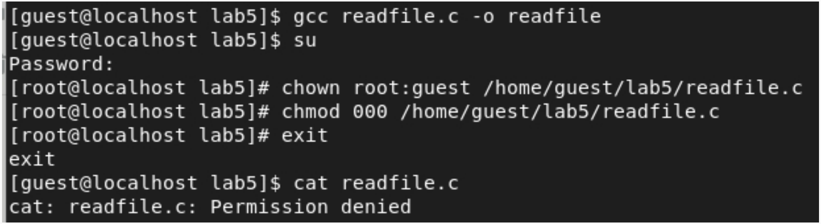
Результат выполнения лабораторной работы



```c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/stat.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 6: рис 6. Запуск readfile.c

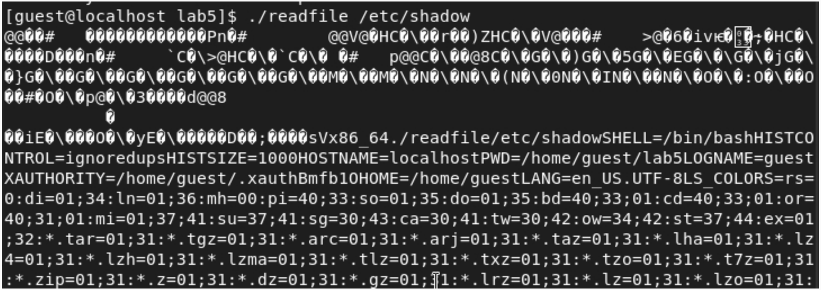
Результат выполнения лабораторной работы



```
[guest@localhost lab5]$ gcc readfile.c -o readfile
[guest@localhost lab5]$ su
Password:
[root@localhost lab5]# chown root:guest /home/guest/lab5/readfile.c
[root@localhost lab5]# chmod 000 /home/guest/lab5/readfile.c
[root@localhost lab5]# exit
exit
[guest@localhost lab5]$ cat readfile.c
cat: readfile.c: Permission denied
```

Figure 7: рис 7. Запуск readfile.c

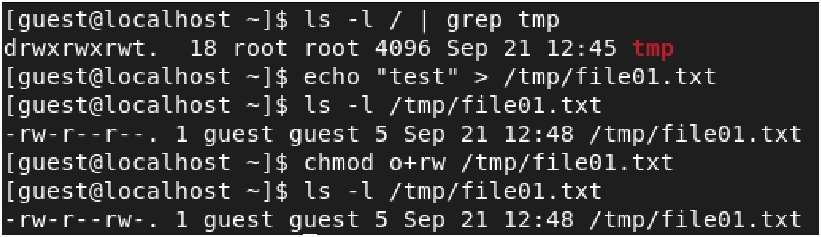
Результат выполнения лабораторной работы



```
[guest@localhost lab5]$ ./readfile /etc/shadow
@@��#   ������������Pn�#        @@V@�HC�\��r��)ZHC�\�V@���#    >@�6�ivє��;�HC�\
����D���n�#     `C�\>@HC�\�`C�\� �#   p@@C�\��@8C�\�G�\�)G�\�5G�\�EG�\�\G�\�jG�\
�}G�\��G�\��G�\��G�\��G�\��G�\��M�\��M�\�N�\�N�\�(N�\�0N�\�IN�\��N�\�O�\�:O�\��O
��#�O�\�p@�\�3����d@@8
          �
��iE�\���O�\�yE�\�����D��;����sVx86_64./readfile/etc/shadowSHELL=/bin/bashHISTCO
NTROL=ignoredupsHISTSIZE=1000HOSTNAME=localhostPWD=/home/guest/lab5LOGNAME=guest
XAUTHORITY=/home/guest/.xauthBmfb1OHOME=/home/guestLANG=en_US.UTF-8LS_COLORS=rs=
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:or=
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:*.lz
4=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;31:
```

Figure 8: рис 8. Проверка readfile.c

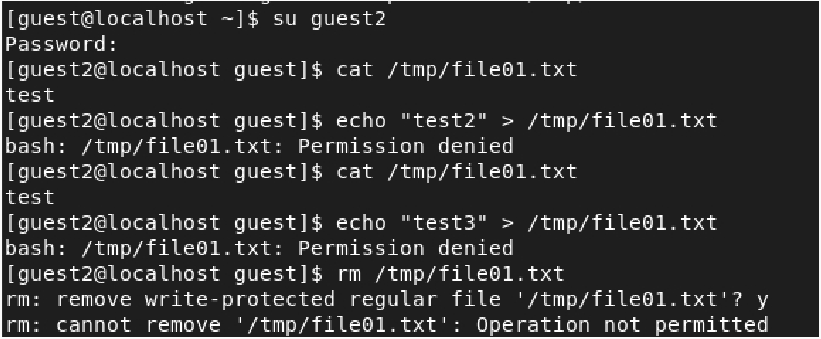
Результат выполнения лабораторной работы



```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt.  18 root root 4096 Sep 21 12:45 tmp
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-r--r--. 1 guest guest 5 Sep 21 12:48 /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 Sep 21 12:48 /tmp/file01.txt
```

Figure 9: рис 9. Тестовый файл

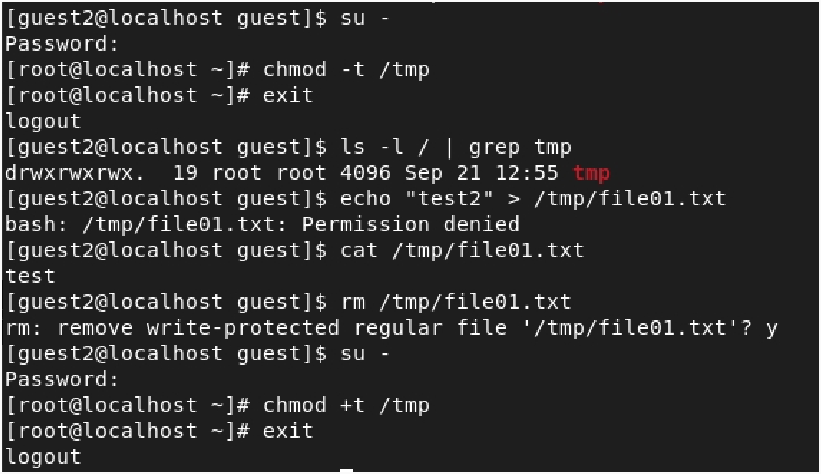
Результат выполнения лабораторной работы



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

Figure 10: рис 10. Изменение файла другим пользователем

Результат выполнения лабораторной работы



```
[guest2@localhost guest]$ su -
Password:
[root@localhost ~]# chmod -t /tmp
[root@localhost ~]# exit
logout
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx.  19 root root 4096 Sep 21 12:55 tmp
[guest2@localhost guest]$ echo "test2" > /tmp/file01.txt
bash: /tmp/file01.txt: Permission denied
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: remove write-protected regular file '/tmp/file01.txt'? y
[guest2@localhost guest]$ su -
Password:
[root@localhost ~]# chmod +t /tmp
[root@localhost ~]# exit
logout
```

Figure 11: рис 11. Снятие атрибута

Вывод

В ходе работы мы изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.