

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

1.С помощью команды gcc -v убедилась, что у меня установлен компилятор gcc (рис.1).

```
[guest@localhost ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/11/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --enable-bootstrap --enable-host-pie --enable-host-bind-now --enable-languages=c,c++,fortran,lto --prefix=/usr --mandir=/usr/share/man --infodir=/usr/share/info --with-bugurl=https://bugs.rockylinux.org/ --enable-shared --enable-threads=posix --enable-checking=release --enable-multi-lib --with-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-unique-object --enable-linker-build-id --with-gcc-major-version-only --with-linker-hash-style=gnu --enable-plugin --enable-initfini-array --without-isl --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
Модель многопоточности: posix
Supported LTO compression algorithms: zlib zstd
gcc версия 11.2.1 20220127 (Red Hat 11.2.1-9) (GCC)
[guest@localhost ~]$
```

рис.1

2.Вошла в систему от имени пользователя guest. Создала программу simpleid.c (рис.2-3).

```
[guest@localhost ~]$ touch simpleid.c
[guest@localhost ~]$ ls
simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
file10     Видео      Загрузки     Музыка         'Рабочий стол'
[guest@localhost ~]$
```

рис.2

```
Открыть  +  *simpleid.c  Сохранить  ☰  ✕
1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main()
7 {
8     uid_t uid = geteuid();
9     gid_t gid = getegid();
10    printf ("uid=%d, gid=%d\n", uid, gid);
11    return 0;
12 }
```

рис.3

3.Скомпилировала программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid (рис.4).

```
[guest@localhost ~]$ gcc simpleid.c -o simpleid
[guest@localhost ~]$ ls
dir1  simpleid  Видео  Загрузки  Музыка  'Рабочий стол'
file10  simpleid.c  Документы  Изображения  Общедоступные  Шаблоны
[guest@localhost ~]$
```

рис.4

4.Выполнила программу simpleid: ./simpleid (рис.5).

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

рис.5

5.Выполнила системную программу id: id (рис.6).

```
[guest@localhost ~]$ id
uid=1001(guest) gid=1001(guest) группы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@localhost ~]$
```

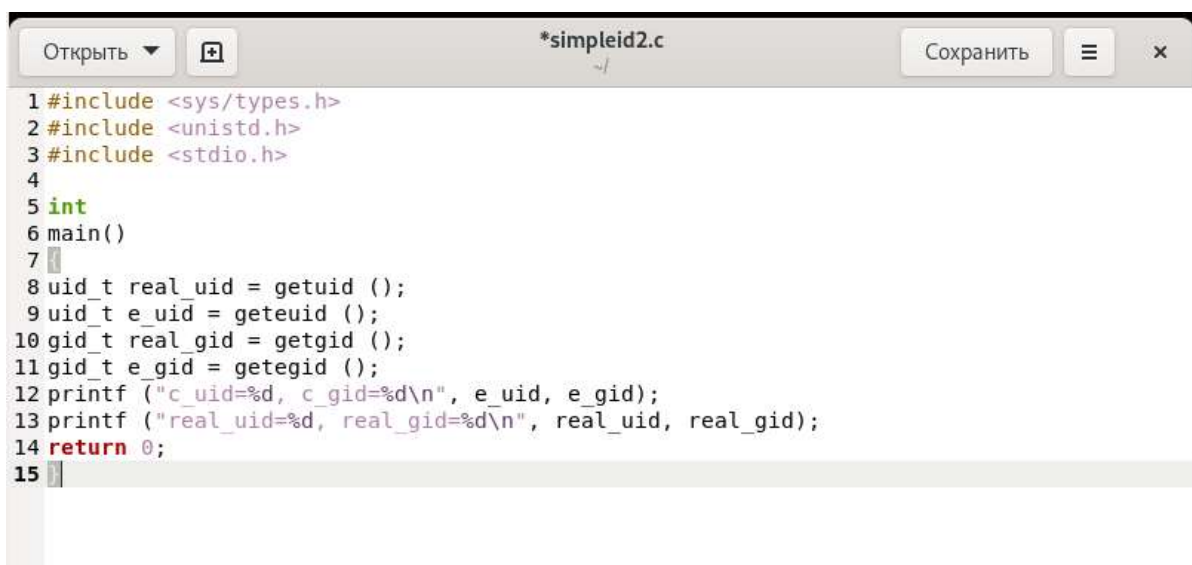
рис.6

Полученный мною результат совпал с данными предыдущего пункта задания.

6.Усложнила программу, добавив вывод действительных идентификаторов. Получившуюся программу назвала simpleid2.c (рис.7-8).

```
[guest@localhost ~]$ touch simpleid2.c
[guest@localhost ~]$
```

рис.7



```
*simpleid2.c
~/
Открыть  +  Сохранить  ≡  ×

1 #include <sys/types.h>
2 #include <unistd.h>
3 #include <stdio.h>
4
5 int
6 main()
7 {
8     uid_t real_uid = getuid ();
9     uid_t e_uid = geteuid ();
10    gid_t real_gid = getgid ();
11    gid_t e_gid = getegid ();
12    printf ("c_uid=%d, c_gid=%d\n", e_uid, e_gid);
13    printf ("real_uid=%d, real_gid=%d\n", real_uid, real_gid);
14    return 0;
15 }
```

рис. 8

7.Скомпилировала и запустила simpleid2.c:

```
gcc simpleid2.c -o simpleid2
```

```
./simpleid2
```

(рис.9)

```
[guest@localhost ~]$ gcc simpleid2.c -o simpleid2
[guest@localhost ~]$ ls
file10      simpleid2  Видео      Изображения  'Рабочий стол'
simpleid     simpleid2.c  Документы  Музыка        Шаблоны
simpleid     simpleid.c  Загрузки   Общедоступные
[guest@localhost ~]$ ./simpleid2
c_uid=1001, c_gid=1001
real_uid=1001, real_gid=1001
[guest@localhost ~]$
```

рис.9

8.От имени суперпользователя выполнила команды:

```
chown root:guest /home/guest/simpleid2
```

```
chmod u+s /home/guest/simpleid2
```

(рис.10-11)

```
[guest@localhost ~]$ su
Пароль:
[root@localhost guest]# chown root:guest /home/guest/simpleid2
```

рис.10

```
[root@localhost guest]# chmod u+s /home/guest/simpleid2
[root@localhost guest]#
```

рис.11

Команда `chown root:guest /home/guest/simpleid2` меняет владельца файла.

Команда `chmod u+s /home/guest/simpleid2` меняет права доступа к файлу.

9.Выполнила проверку правильности установки новых атрибутов и смены владельца файла `simpleid2`: `ls -l simpleid2` (рис.12).

```
[root@localhost guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 26008 окт  6 15:13 simpleid2
[root@localhost guest]#
```

рис.12

10.Запустила `simpleid2` и `id`:

```
./simpleid2
```

```
id
```

(рис.13)

```
[root@localhost guest]# ./simpleid2
c_uid=0, c_gid=0
real_uid=0, real_gid=0
[root@localhost guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost guest]#
```

рис.13

В отличие от предыдущих результатов, я получила значения 0.

11.Прodelала тоже самое относительно SetGID-бита (рис.14-16).

```
[root@localhost guest]# chmod u-s /home/guest/simpleid2
[root@localhost guest]# chmod g+s /home/guest/simpleid2
[root@localhost guest]#
```

рис.14

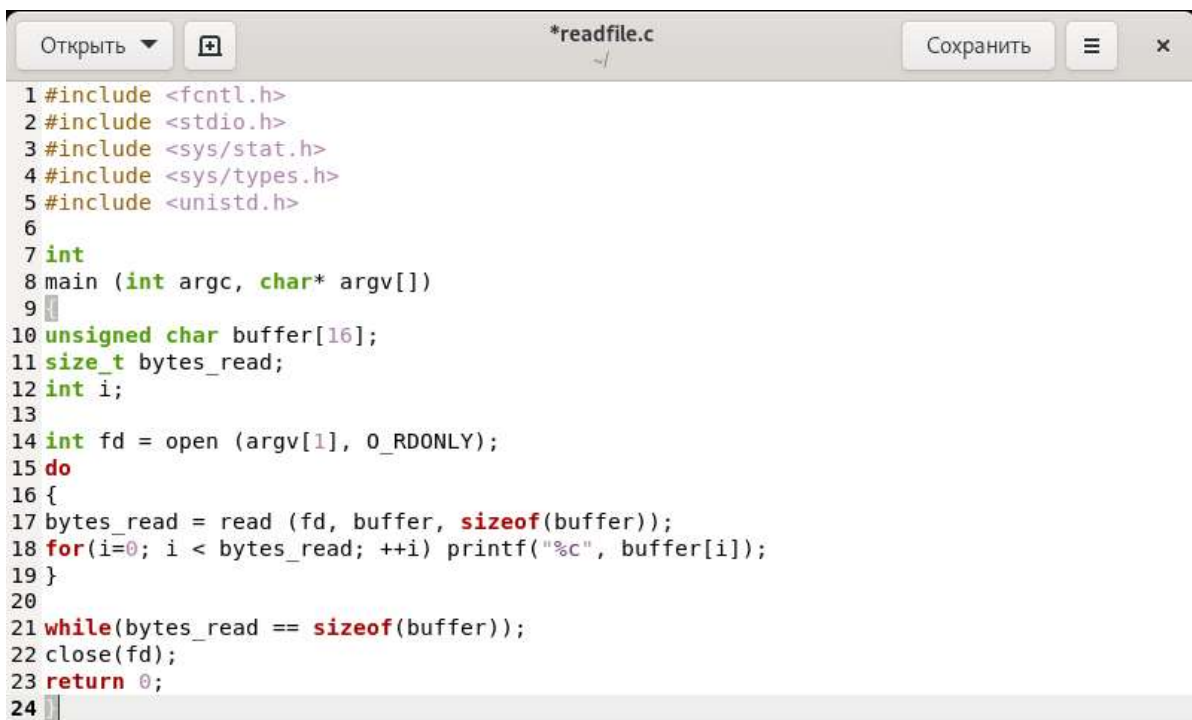
```
[root@localhost guest]# ls -l simpleid2
-rwxrwsr-x. 1 root guest 26008 окт  6 15:13 simpleid2
[root@localhost guest]#
```

рис.15

```
[root@localhost guest]# ./simpleid2
c_uid=0, c_gid=1001
real_uid=0, real_gid=0
[root@localhost guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost guest]#
```

рис.16

12.Создала программу readfile.c (рис.17).



```
Открыть + *readfile.c ~/ Сохранить ☰ ×
1 #include <fcntl.h>
2 #include <stdio.h>
3 #include <sys/stat.h>
4 #include <sys/types.h>
5 #include <unistd.h>
6
7 int
8 main (int argc, char* argv[])
9 {
10 unsigned char buffer[16];
11 size_t bytes_read;
12 int i;
13
14 int fd = open (argv[1], O_RDONLY);
15 do
16 {
17 bytes_read = read (fd, buffer, sizeof(buffer));
18 for(i=0; i < bytes_read; ++i) printf("%c", buffer[i]);
19 }
20
21 while(bytes_read == sizeof(buffer));
22 close(fd);
23 return 0;
24 }
```

рис.17

13.Откомпилировала её: gcc readfile.c -o readfile (рис.18).

```
[root@localhost guest]# gcc readfile.c -o readfile
[root@localhost guest]#
```

рис.18

14.Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, а guest не мог (рис.19).

```
[root@localhost guest]# chown root:guest /home/guest/readfile.c
[root@localhost guest]# chmod 700 /home/guest/readfile.c
[root@localhost guest]#
```


рис.19

15.Проверила, что пользователь guest не может прочитать файл readfile.c (рис.20).

```
[guest@localhost ~]$ ls -l readfile.c
-rwx-----. 1 root guest 399 окт  6 15:53 readfile.c
[guest@localhost ~]$ cat readfile.c
cat: readfile.c: Отказано в доступе
[guest@localhost ~]$
```

рис.20

16.Сменила у программы readfile владельца и установила SetUID-бит (рис.21).

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

рис.21

17.Проверила, может ли программа readfile прочитать файл readfile.c (рис.22).

```
[guest@localhost ~]$ ./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@localhost ~]$
```

рис.22

18.Проверила, может ли программа readfile прочитать файл /etc/shadow (рис.23-24).

```
[guest@localhost ~]$ ./readfile /etc/shadow
root:$6$/S28hjzry5M0emR$g5nduZ6uDqgXC6Vn6GTgKNXaZsgNUG27Ws4wUMMtdZ4Y3jpoGUMhpEnVmXhiUT
pIFDl6GQKi.Z1bQopvkNKCf1::0:99999:7:::
bin:!:19123:0:99999:7:::
daemon:!:19123:0:99999:7:::
adm:!:19123:0:99999:7:::
lp:!:19123:0:99999:7:::
sync:!:19123:0:99999:7:::
shutdown:!:19123:0:99999:7:::
halt:!:19123:0:99999:7:::
mail:!:19123:0:99999:7:::
operator:!:19123:0:99999:7:::
games:!:19123:0:99999:7:::
ftp:!:19123:0:99999:7:::
nobody:!:19123:0:99999:7:::
systemd-coredump:!!:19249:::::::
dbus:!!:19249:::::::
polkitd:!!:19249:::::::
rtkit:!!:19249:::::::
sssd:!!:19249:::::::
avahi:!!:19249:::::::
pipewire:!!:19249:::::::
libstoragemgmt:!!:19249:::::::
tss:!!:19249:::::::
geoclue:!!:19249:::::::
cockpit-ws:!!:19249:::::::
cockpit-wsinstance:!!:19249:::::::
setroubleshoot:!!:19249:::::::
flatpak:!!:19249:::::::
colord:!!:19249:::::::
clevis:!!:19249:::::::
```

Актив
Чтобы ак

рис.23

```
clevis:!!:19249:::::::
gdm:!!:19249:::::::
systemd-oom:!:19249:::::::
pesign:!!:19249:::::::
gnome-initial-setup:!!:19249:::::::
sshd:!!:19249:::::::
chrony:!!:19249:::::::
dnsmasq:!!:19249:::::::
tcpdump:!!:19249:::::::
oyaaseeva:$6$0t$qLom4fomJM6dL1Z2ibucp/V0cQqAXbjVdxiF1M00LnzX4cNMgUNziXEyTCzLPjY1HRuQMD7
Iy2AdHwMUYxu0b0:19249:0:99999:7:::
vboxadd:!!:19249:::::::
guest:$6$0rXuuNTPC.yP8147$jSSwo8hmM1rMQNh8Zr6Tqd0lad2TlIEJAIAxtpN9dq5aEQdxyPLEGT2udhLNN
K92c0LmxmK7R1BaQEXTFo6Wy.:19251:0:99999:7:::
guest2:$6$uA.9TeTSU0j//ro0$Iew36RBrsFkHKgv0uZ5cDdPLxdumHpwIloxZad0B.cZFIfc0zQd4QP0bNrzi
dUSHFry008SJ4d3luCxt4JIhN1:19255:0:99999:7:::
[guest@localhost ~]$ █
```

Актив
Чтобы ак

рис.24

Так как у программы установлен SetUID-бит, то ей временно предоставляются права владельца файла.

19.Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду `ls -l / | grep tmp` (рис.25).

```
[guest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 окт  6 15:53 tmp
[guest@localhost ~]$
```

рис.25

20.От имени пользователя guest создала файл file01.txt в директории /tmp со словом test: `echo "test" > /tmp/file01.txt` (рис.26).

```
[guest@localhost ~]$ echo "test" > /tmp/file01.txt
[guest@localhost ~]$ cat /tmp/file01.txt
test
[guest@localhost ~]$
```

рис.26

21.Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»:

```
ls -l /tmp/file01.txt
```

```
chmod o+rw /tmp/file01.txt
```

```
ls -l /tmp/file01.txt
```

(рис.27)

```
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 guest guest 5 окт  6 16:08 /tmp/file01.txt
[guest@localhost ~]$ chmod o+rw /tmp/file01.txt
[guest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 guest guest 5 окт  6 16:08 /tmp/file01.txt
[guest@localhost ~]$
```

рис.27

22.От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt: cat /tmp/file01.txt (рис.28).

```
[guest@localhost ~]$ su guest2
Пароль:
[guest2@localhost guest]$ cat /tmp/file01.txt
test
[guest2@localhost guest]$
```

рис.28

23.От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt (рис.29).

```
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$
```

рис.29

Операцию удалось выполнить.

24.Проверила содержимое файла командой cat /tmp/file01.txt (рис.30).

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test
test2
[guest2@localhost guest]$
```

рис.30

25.От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo "test3" > /tmp/file01.txt (рис.31).

```
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$
```

рис.31

Операцию удалось выполнить.

26.Проверила содержимое файла командой cat /tmp/file01.txt (рис.32).

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$
```

рис.32

27.От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой rm /tmp/file01.txt

(рис.33).

```
[guest2@localhost guest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[guest2@localhost guest]$
```

рис.33

Операцию выполнить не удалось.

28.Повысила свои права до суперпользователя командой su и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис.34).

```
[guest2@localhost guest]$ su
Пароль:
[root@localhost guest]# chmod -t /tmp
[root@localhost guest]#
```

рис.34

29.Покинула режим суперпользователя командой exit (рис.35).

```
[root@localhost guest]# exit
exit
[guest2@localhost guest]$
```

рис.35

30.От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис.36).

```
[guest2@localhost guest]$ ls -l / | grep tmp
drwxrwxrwx. 16 root root 4096 окт  6 16:39 tmp
[guest2@localhost guest]$
```

рис.36

31.Повторила предыдущие шаги (рис.37).

```
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ echo "test2" >> /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
test2
[guest2@localhost guest]$ echo "test3" > /tmp/file01.txt
[guest2@localhost guest]$ cat /tmp/file01.txt
test3
[guest2@localhost guest]$ rm /tmp/file01.txt
[guest2@localhost guest]$ ls
tmp      readfile.c  simpleid2.c  Документы  Музыка      Шаблоны
file10   simpleid    simpleid.c   Загрузки   Общедоступные
readfile simpleid2    Видео       Изображения 'Рабочий стол'
[guest2@localhost guest]$ ls /tmp
dbus-bafaldJrjs
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-chrond.service-YSiWEI
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-colord.service-Ph9aty
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-dbus-broker.service-hufUky
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-fwupd.service-KpwLjk
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-ModemManager.service-8vKuCu
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-power-profiles-daemon.service-3hb8zw
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-rtkit-daemon.service-4Kwg5P
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-switcheroo-control.service-o8NhF8
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-systemd-logind.service-wKEI0F
systemd-private-79be8f3b3ec94da99df56f5fa7f12ed2-upower.service-38FquK
[guest2@localhost guest]$
```

рис.37

Я смогла удалить файл от имени пользователя, не являющегося его владельцем. Sticky-bit позволяет защищать файлы от случайного удаления, когда несколько пользователей имеют права на запись в один и тот же каталог. Если у файла атрибут t стоит, значит пользователь может удалить файл, только если он является пользователем-владельцем файла или каталога, в котором содержится файл. Если же этот атрибут не установлен, то удалить файл могут все пользователи, которым позволено удалять файлы из каталога.

32.Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp:

su

chmod +t /tmp

exit

(рис.38)

```
[guest2@localhost guest]$ su
Пароль:
[root@localhost guest]# chmod +t /tmp
[root@localhost guest]# ls -l / | grep tmp
drwxrwxrwt. 16 root root 4096 окт  6 16:50 tmp
[root@localhost guest]# exit
exit
[guest2@localhost guest]$
```

рис.38

Выводы

В ходе выполнения лабораторной работы я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Я получила практические навыки работы в консоли с дополнительными атрибутами. Я рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Лабораторная работа №5.