# presentation

# Презентация по лабораторной работе №5

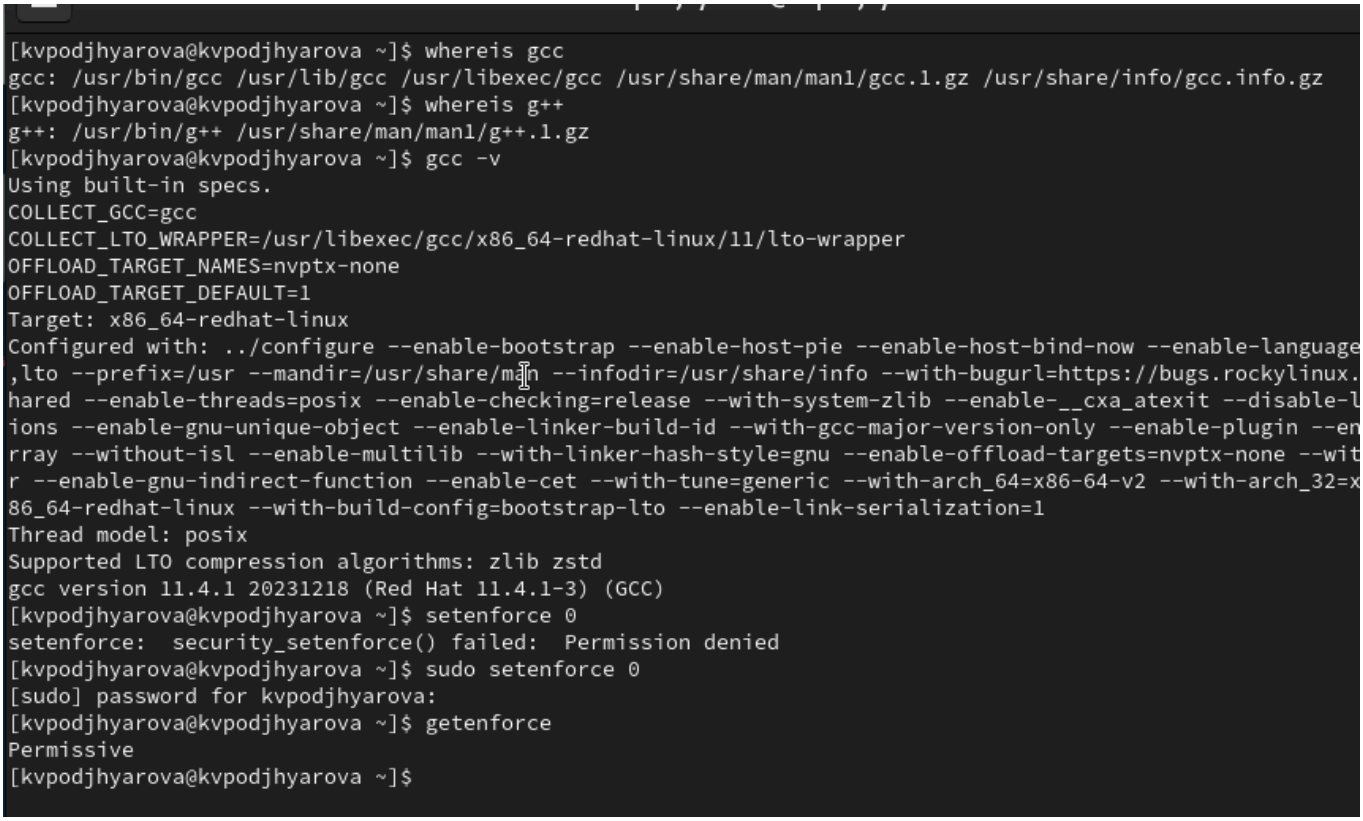
## Подъярова Ксения Витальевна

# Цель работы

Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

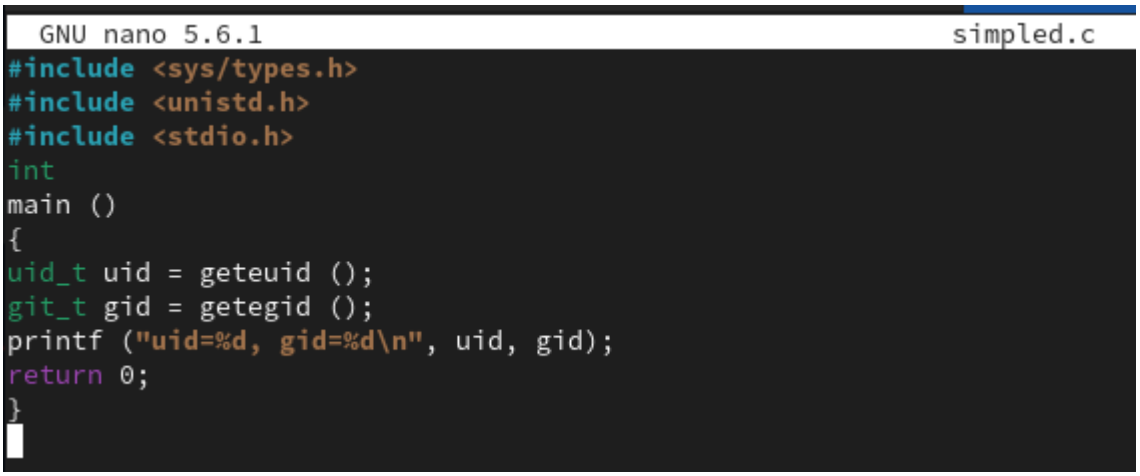
# Выполнение лабораторной работы

1.Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, комнда gcc -v позволяет это сделать. Также осуществляется отключение системы запретом с помощью setenforce 0 (рис. 1).



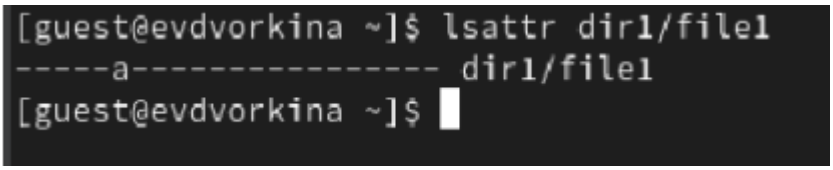

2. Осуществляется вход от имени пользователя guest

3. Создание файла simpled.c и запись в файл кода. C++ Листинг 1 #include #include #include int main () { uid_t uid = geteuid (); gid_t gid = getegid (); printf ("uid=%d, gid=%d\n", uid, gid); return 0; } Содержимое файла выглядит следующим образом (рис. 3).

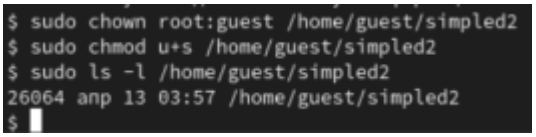



4. Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп, от вывода при вводе if, они отличаются только тем, что информации меньше

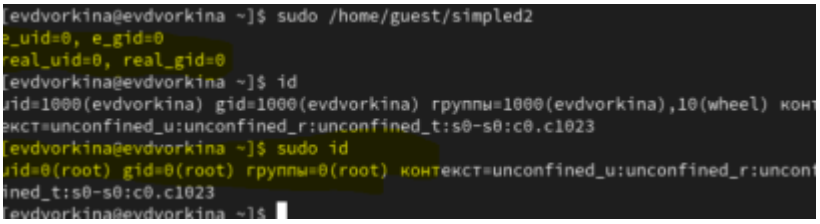

5. Создание, запись в файл и компиляция файла simpled2.c. Запуск программы




6. С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа




7. Сравнение вывода программы и команды id, наша команда снова вывела только ограниченное количество информации



Создание и компиляция файла readfile.c

```
touch readfile.c
nano readfile.c
nano readfile.c
gcc readfile.c -o readfile
ls
```

```
 GNU nano 5.6.1                    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;
}
```

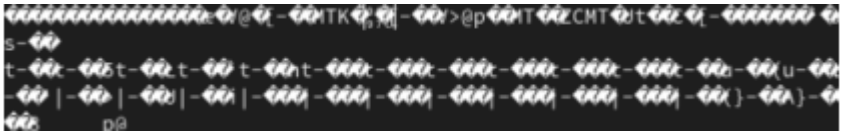
8. Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла

```
sudo chown root:guest /home/guest/readfile.c
sudo chmod u+s /home/guest/readfile.c
sudo chmod 700 /home/guest/readfile.c
sudo chmod -r /home/guest/readfile.c
sudo chmod u+s /home/guest/readfile.c
```

9. Проверка прочесть файл от имени пользователя guest. Прочесть файл не удается

```
cat: readfile.c: Отказано в доступе
```

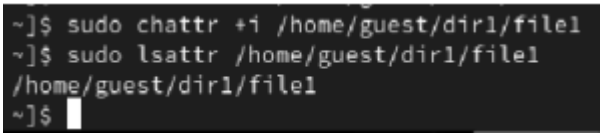
Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем “отказано в доступе". Попытка прочесть файл \etc\shadow с помощью программы, все еще получаем отказ в доступе.



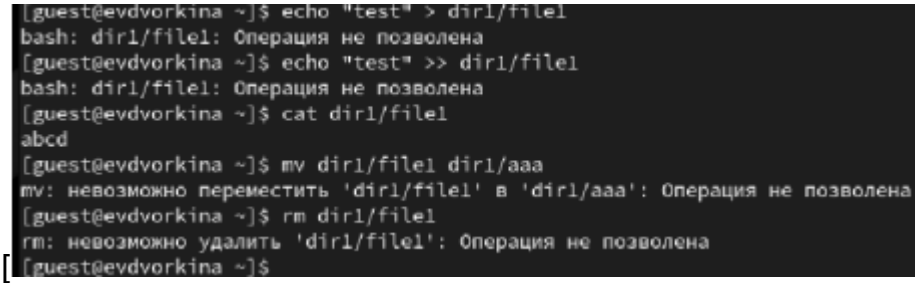
10. Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно

```
root:$6$3reywnbOG.0EfHL7$1td/ZD0qRQQEdDaZehnNr0Kq7lhY9Hs4IpOCdU6M/hMkBvHfSqsO2
gd3/YkGPNmw5AD2t0THlFZYuXi4eD/rU0::0:99999:7:::
bin:*:19469:0:99999:7:::
daemon:*:19469:0:99999:7:::
adm:*:19469:0:99999:7:::
lp:*:19469:0:99999:7:::
```

Добавляю расширенный атрибут i от имени суперпользователя, теперь все выполнено верно (рис. 13).




Пытаюсь записать в файл, дозаписать, переименовать или удалить, ничего из этого сделать нельзя (рис. 14).



![

# Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky□биты. Получила практические навыки работы в кон- соли с дополнительными атрибутами. Рассмотрела работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.