

# Отчёт по лабораторной работе 5

---

Радимов Игорь

## 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.

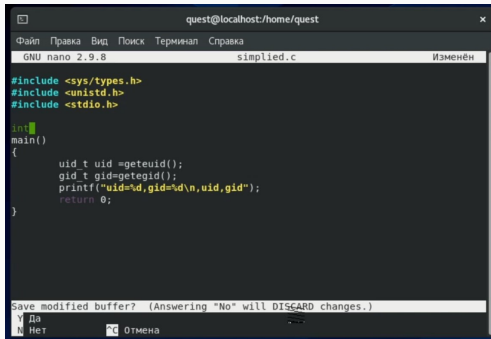
# Задание

---

Лабораторная работа подразумевает изучение влияния дополнительных атрибутов на файлы пользователя и изучение механизмов изменения идентификаторов.

# Выполнение основной части лабораторной работы

1. Создал программу simpleid.c (рис. 1).



```
quest@localhost:/home/quest
GNU nano 2.9.8      simplified.c      Изменён

#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t uid =getuid();
    gid_t gid=getegid();
    printf("uid=%d,gid=%d\n,uid,gid");
    return 0;
}

Save modified buffer? (Answering "No" will DISCARD changes.)
Да
Нет      Отмена
```

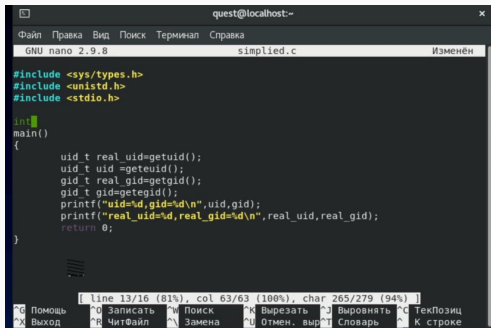
**Figure 1:** рис.1. Программа simpleid.c

2. Выполнил системную программу `id` и сравнил полученный результат с данными предыдущего пункта задания (рис. 2). Видим, что пользователи и группы совпадают. При этом команда `id` вывела действительные идентификаторы, а программа вывел эффективные, но при этом они совпадают и выводят 1001, то есть пользователя `quest`.

```
[quest@localhost ~]$ su
Пароль:
[root@localhost quest]# nano simplified.c
[root@localhost quest]# su quest
[quest@localhost ~]$ gcc simplified.c -o simplified
[quest@localhost ~]$ ./simplified
uid=1001,gid=1001
[quest@localhost ~]$ id
uid=1001(quest) gid=1001(quest) rpyнны=1001(quest) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

**Figure 2:** рис.2. Компиляция программы `simpleid`, её выполнение и сравнение с командой `id`.

3. Усложнил программу, добавив вывод действительных идентификаторов, получившуюся программу назвал `simplified2.c` (рис. 3).



```
quest@localhost:~$ nano simplified.c
GNU nano 2.9.8 simplified.c
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

int
main()
{
    uid_t real_uid=getuid();
    uid_t uid =geteuid();
    gid_t real_gid=getgid();
    gid_t gid=getegid();
    printf("uid=%d,gid=%d\n",uid,gid);
    printf("real_uid=%d,real_gid=%d\n",real_uid,real_gid);
    return 0;
}
```

**Figure 3:** рис.3. Программа `simplified2.c`



4. Запустил `simpleid2` и `id` командами `./simpleid2` и `id` (рис. 4). Сравнил результаты: действительные идентификаторы совпадают с выводом команды `id` - везде 0, то есть рут-пользователь. Так же важно заметить, что эффективные идентификаторы совпадают с действительными.

```
[quest@localhost ~]$ su
Пароль:
[root@localhost quest]# chown root:quest /home/quest/simplified
[root@localhost quest]# chmod u+s /home/quest/simplified
[root@localhost quest]# ls -l simplified
-rwsrwxr-x. 1 root quest 17648 ноя  1 14:19 simplified
[root@localhost quest]# ./simplified
uid=0,gid=0
real uid=0,real_gid=0
[root@localhost quest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

**Figure 4:** рис.4. Изменение владельца программы и установка SetUID-бита, проверка установки и изменения, запуск программы и команды `id`.

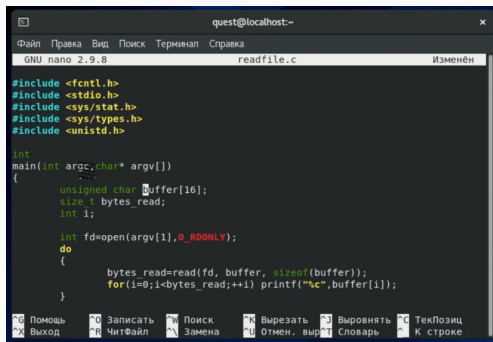
## 5. Проделал тоже самое относительно SetGID-бита (рис. 5)

Установка SetGID-бита отражается к команде ls, а сравнение выполнения программы и команды id дало следующие результаты: действительные идентификаторы совпадают с выводом команды id - везде 0, то есть рут-пользователь. Но так же важно заметить, что эффективные идентификаторы отличны от действительных: пользователь - 0, группа - 1001.

```
ined t:s0-s0:c0.c1023
[root@localhost quest]# chmod g+s /home/quest/simplified
[root@localhost quest]# ls -l simplified
-rwsrwsr-x. 1 root quest 17648 ноя  1 14:19 simplified
[root@localhost quest]# ./simplified
uid=0,gid=1001
real uid=0,real gid=0
[root@localhost quest]# id
uid=0(root) gid=0(root) rpyнпы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@localhost quest]#
```

**Figure 5:** рис.5. Установка SetGID-бита, проверка установки, запуск программы и команды id.

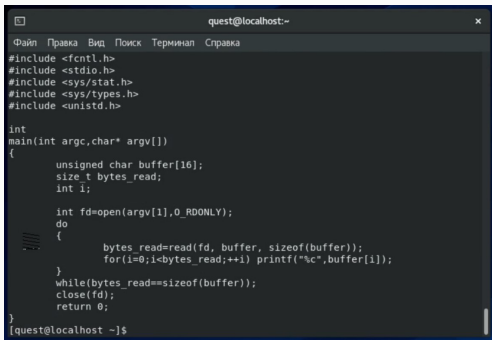
## 6. Создал программу readfile.c (рис. 6).



```
quest@localhost:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
GNU nano 2.9.8      readfile.c      Изменён  
  
#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]);  
    }  
}
```

**Figure 6:** рис.6. Программа readfile.c

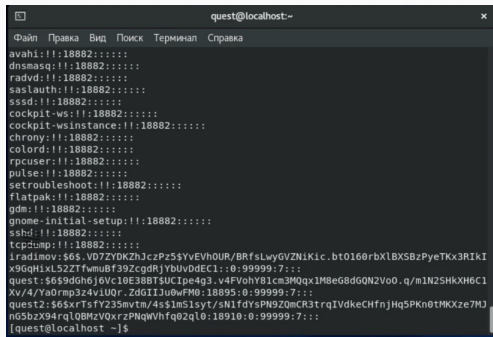
7. Проверил, может ли программа readfile прочитать файл readfile.c. Да, может (рис. 7).



```
quest@localhost:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
#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;  
}  
[quest@localhost ~]$
```

**Figure 7:** рис.7. Выполнение программы readfile с файлом readfile.c.

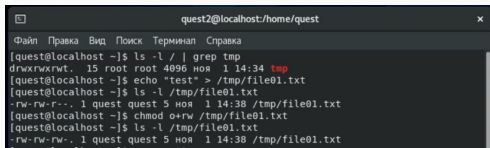
8. Проверил, может ли программа readfile прочитать файл /etc/shadow. Её выполнению возможно в том числе, так как владельцем файла является root-пользователь (рис. 8).



```
quest@localhost:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
avahi:!!:18882:::~  
dnsmasq:!!:18882:::~  
radvd:!!:18882:::~  
ssslauth:!!:18882:::~  
sssd:!!:18882:::~  
cockpit-ws:!!:18882:::~  
cockpit-wsinstance:!!:18882:::~  
chrony:!!:18882:::~  
colord:!!:18882:::~  
rpcuser:!!:18882:::~  
pulse:!!:18882:::~  
setroubleshoot:!!:18882:::~  
flatpak:!!:18882:::~  
gdm:!!:18882:::~  
gnome-initial-setup:!!:18882:::~  
sshd:!!:18882:::~  
tcpdump:!!:18882:::~  
iradimov:$6$.V07ZYDKzhJczPz5$YvEVh0UR/BRfsLwyGVZniKic.bt0160rbXlBXSBzPyeTKx3RIkI  
x9GqHixL52ZTfwmBf39ZcgdrJYbUvDdEC1::0:99999:7:::  
quest:$6$9dGh6j6Vc10E38BT$UCIpe4g3.v4FVohY81cm3MQqx1M8eG8dGQn2Vo0.q/m1N2SHKxH6C1  
Xv/4/Ya0rmp3z4vIUOr.ZdGIIJu0wFM0:18895:0:99999:7:::  
quest2:$6$xrTsFY235mvtm/4s$1m51syT/sN1fdYsPN9ZQmCR3trqIVdkeCHfnjHq5PKn0tMKXze7MJ  
nG5bzX94rql08MzVQxrzPNqWVhfq02ql0:18910:0:99999:7:::  
[quest@localhost ~]$
```

**Figure 8:** рис.8. Выполнение программы readfile с файлом /etc/shadow.

9. От пользователя quest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt командой `cat /tmp/file01.txt` (рис. 9).



```
quest2@localhost/home/quest
Файл Правка Вид Поиск Терминал Справка
[quest@localhost ~]$ ls -l / | grep tmp
drwxrwxrwt. 15 root root 4096 ноя  1 14:34 tmp
[quest@localhost ~]$ echo "test" > /tmp/file01.txt
[quest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-r--. 1 quest quest 5 ноя  1 14:38 /tmp/file01.txt
[quest@localhost ~]$ chmod o+rw /tmp/file01.txt
[quest@localhost ~]$ ls -l /tmp/file01.txt
-rw-rw-rw-. 1 quest quest 5 ноя  1 14:38 /tmp/file01.txt
```

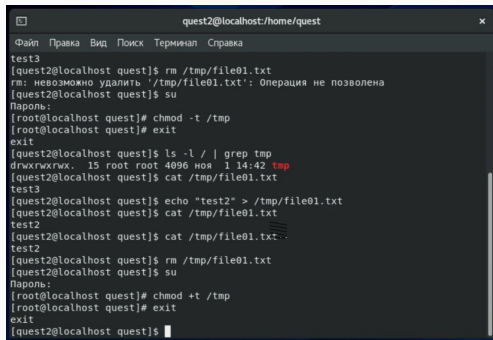
**Figure 9:** рис.9. Выполнение пунктов 1-4 исследования Sticky-бита

10. От пользователя quest2 попробовал удалить файл /tmp/file01.txt командой `rm /tmp/file01.txt`. Мне не удалось удалить файл (рис. 10).

```
[quest@localhost ~]$ su quest2
Пароль:
[quest2@localhost quest]$ cat /tmp/file01.txt
test
[quest2@localhost quest]$ echo "test2" > /tmp/file01.txt
[quest2@localhost quest]$ cat /tmp/file01.txt
test2
[quest2@localhost quest]$ echo "test3" > /tmp/file01.txt
[quest2@localhost quest]$ cat /tmp/file01.txt
test3
[quest2@localhost quest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': Операция не позволена
[quest2@localhost quest]$ su
Пароль:
```

**Figure 10:** рис.10. Выполнение пунктов 5-9 исследования Sticky-бита .

11. Повторил предыдущие шаги (рис. 11). Видим, что дозапись и запись так же разрешены, но при этом удалось и удалить файл. Мне удалось удалить файл от имени пользователя, не являющегося его владельцем.



```
quest2@localhost/home/quest
Файл  Правка  Вид  Поиск  Терминал  Справка
test3
[quest2@localhost quest]$ rm /tmp/file01.txt
rm: невозможно удалить '/tmp/file01.txt': операция не позволена
[quest2@localhost quest]$ su
Пароль:
[root@localhost quest]# chmod -t /tmp
[root@localhost quest]# exit
exit
[quest2@localhost quest]$ ls -l / | grep tmp
drwxrwxrwx. 15 root root 4096 ноя  1 14:42 tmp
[quest2@localhost quest]$ cat /tmp/file01.txt
test3
[quest2@localhost quest]$ echo "test2" > /tmp/file01.txt
[quest2@localhost quest]$ cat /tmp/file01.txt
test2
[quest2@localhost quest]$ cat /tmp/file01.txt
test2
[quest2@localhost quest]$ rm /tmp/file01.txt
[quest2@localhost quest]$ su
Пароль:
[root@localhost quest]# chmod +t /tmp
[root@localhost quest]# exit
exit
[quest2@localhost quest]$
```

**Figure 11:** рис.11. Выполнение пунктов 10-13 исследования Sticky-бита .



**Спасибо за внимание**