

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Абузярова Лейла Джамилевна НБИбд-01-19

8 октября, 2022, Москва, Россия

Российский Университет Дружбы Народов

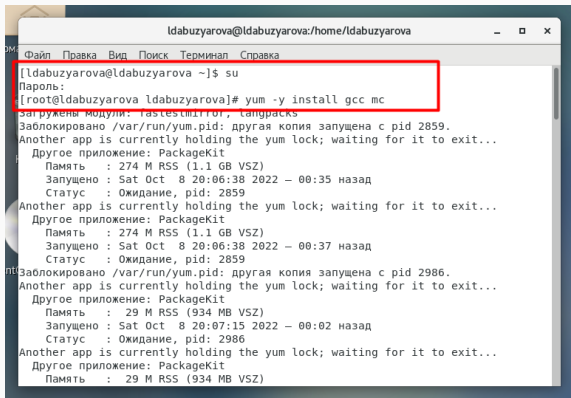
Цели и задачи

Цель 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.

Выполнение лабораторной работы

Подготовка к работе

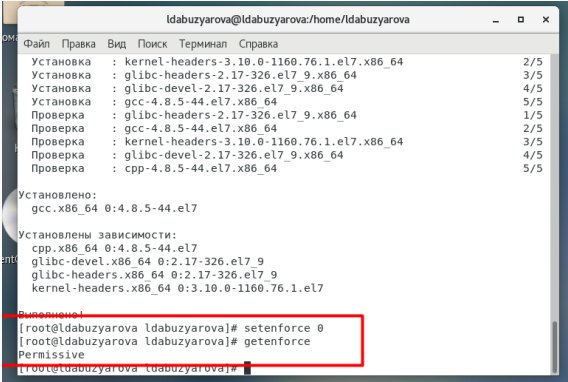


A terminal window titled 'ldabuzyarova@ldabuzyarova:/home/ldabuzyarova' with a menu bar (Файл, Правка, Вид, Поиск, Терминал, Справка). The terminal shows a user switching to root and installing gcc. A red rectangle highlights the command 'yum -y install gcc mc'. The output shows the yum lock being held by another process, which is identified as PackageKit. The process details for PackageKit are shown twice, indicating it was running during the installation attempt.

```
ldabuzyarova@ldabuzyarova ~]$ su
Пароль:
[root@ldabuzyarova ldabuzyarova]# yum -y install gcc mc
Загружены модули: fastestmirror, langpacks
Заблокировано /var/run/yum.pid: другая копия запущена с pid 2859.
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 274 M RSS (1.1 GB VSZ)
Запущено : Sat Oct 8 20:06:38 2022 - 00:35 назад
Статус : Ожидание, pid: 2859
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 274 M RSS (1.1 GB VSZ)
Запущено : Sat Oct 8 20:06:38 2022 - 00:37 назад
Статус : Ожидание, pid: 2859
Заблокировано /var/run/yum.pid: другая копия запущена с pid 2986.
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 29 M RSS (934 MB VSZ)
Запущено : Sat Oct 8 20:07:15 2022 - 00:02 назад
Статус : Ожидание, pid: 2986
Another app is currently holding the yum lock; waiting for it to exit...
Другое приложение: PackageKit
Память : 29 M RSS (934 MB VSZ)
```

Figure 1: Установка компилятора gcc

Подготовка к работе



```
ldabuzyarova@ldabuzyarova:/home/ldabuzyarova
Файл  Правка  Вид  Поиск  Терминал  Справка
Установка : kernel-headers-3.10.0-1160.76.1.el7.x86_64      2/5
Установка : glibc-headers-2.17-326.el7_9.x86_64           3/5
Установка : glibc-devel-2.17-326.el7_9.x86_64             4/5
Установка : gcc-4.8.5-44.el7.x86_64                       5/5
Проверка  : glibc-headers-2.17-326.el7_9.x86_64           1/5
Проверка  : gcc-4.8.5-44.el7.x86_64                       2/5
Проверка  : kernel-headers-3.10.0-1160.76.1.el7.x86_64    3/5
Проверка  : glibc-devel-2.17-326.el7_9.x86_64             4/5
Проверка  : cpp-4.8.5-44.el7.x86_64                       5/5

Установлено:
gcc.x86_64 0:4.8.5-44.el7

Установлены зависимости:
cpp.x86_64 0:4.8.5-44.el7
glibc-devel.x86_64 0:2.17-326.el7_9
glibc-headers.x86_64 0:2.17-326.el7_9
kernel-headers.x86_64 0:3.10.0-1160.76.1.el7

Выполнено!
[root@ldabuzyarova ldabuzyarova]# setenforce 0
[root@ldabuzyarova ldabuzyarova]# getenforce
Permissive
[root@ldabuzyarova ldabuzyarova]#
```

Figure 2: Подготовка к работе

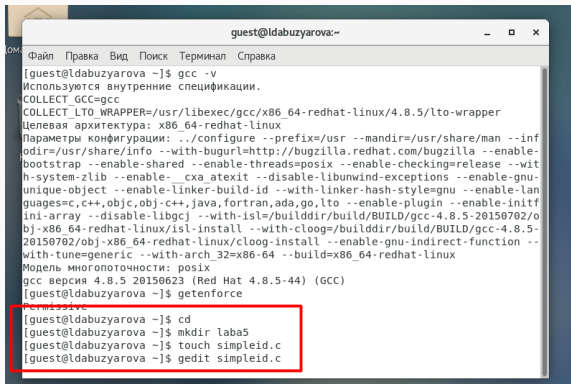
Подготовка к работе

A terminal window titled 'guest@ldabuzyarova:~' with a menu bar (Файл, Правка, Вид, Поиск, Терминал, Справка). The terminal shows the output of 'gcc -v' and 'getenforce'. Two red rectangles highlight the command 'gcc -v' and the output 'Permissive'.

```
guest@ldabuzyarova ~]$ gcc -v
используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-lan
guages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgck --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86_64 --build=x86_64-redhat-linux
Модель многопоточности: posix_
gcc версия 4.8.5-20150623 (Red Hat 4.8.5-44) (GCC)
guest@ldabuzyarova ~]$ getenforce
Permissive
guest@ldabuzyarova ~]$
```

Figure 3: Проверка установки компилятора

Работа с файлом simpleid

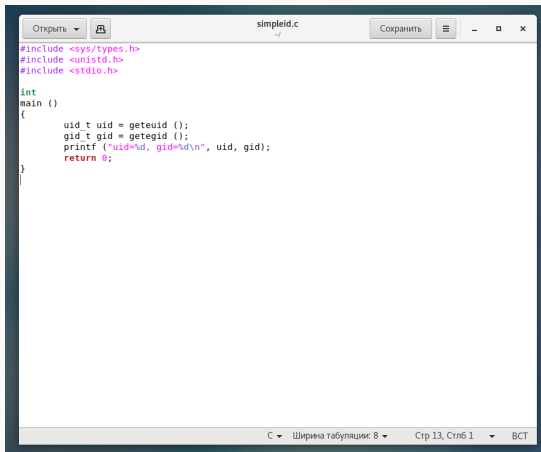


The screenshot shows a terminal window titled "guest@ldabuzyarova:~". The terminal output includes the compilation of simpleid.c using gcc, followed by the creation of the file. The last four lines of the terminal output are highlighted with a red box:

```
[guest@ldabuzyarova ~]$ gcc -v
Используются внутренние спецификации.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/libexec/gcc/x86_64-redhat-linux/4.8.5/lto-wrapper
Целевая архитектура: x86_64-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable- cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-linker-build-id --with-linker-hash-style=gnu --enable-lan
guages=c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgcj --with-isl=/build/buildd/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/build/buildd/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[guest@ldabuzyarova ~]$ getenforce
Permissive
[guest@ldabuzyarova ~]$ cd
[guest@ldabuzyarova ~]$ mkdir laba5
[guest@ldabuzyarova ~]$ touch simpleid.c
[guest@ldabuzyarova ~]$ gedit simpleid.c
```

Figure 4: Создание и редактирование файла simpleid.c

Работа с файлом simpleid



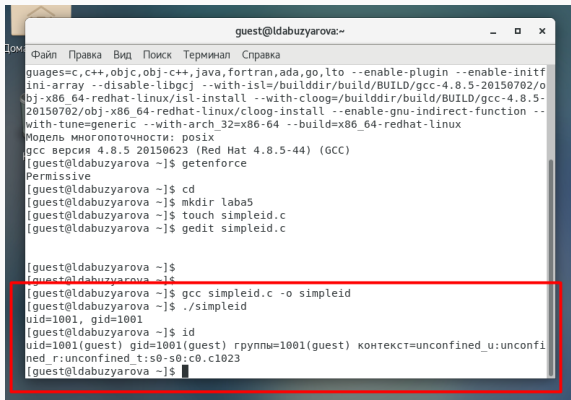
```
#include <sys/types.h>
#include <unistd.h>
#include <stdio.h>

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

The screenshot shows a code editor window with the title 'simpleid.c'. The code is written in C and includes headers for system types, unistd, and stdio. The main function calls geteuid and getegid to retrieve the effective user and group IDs, respectively, and prints them using printf. The status bar at the bottom indicates 'Стр 13, Стлб 1' and 'ВСТ'.

Figure 5: Написание программы

Работа с файлом simpleid



A terminal window titled 'guest@ldabuzyarova:~' with a menu bar containing 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal shows the following commands and output:

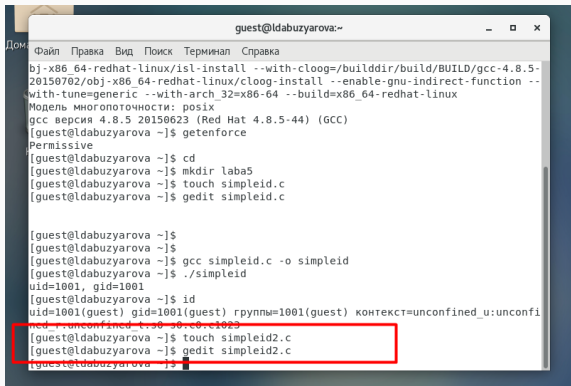
```
languages=c,c++,objc,obj-c++,java,fortran,ada,go,lto --enable-plugin --enable-initf
ini-array --disable-libgcj --with-isl=/builddir/build/BUILD/gcc-4.8.5-20150702/o
bj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-
20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --
with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux
Модель многопоточности: posix
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)
[guest@ldabuzyarova ~]$ getenforce
Permissive
[guest@ldabuzyarova ~]$ cd
[guest@ldabuzyarova ~]$ mkdir laba5
[guest@ldabuzyarova ~]$ touch simpleid.c
[guest@ldabuzyarova ~]$ gedit simpleid.c

[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$ gcc simpleid.c -o simpleid
[guest@ldabuzyarova ~]$ ./simpleid
uid=1001, gid=1001
[guest@ldabuzyarova ~]$ id
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi
ned_r:unconfined_t:s0-s0:c0.c1023
[guest@ldabuzyarova ~]$
```

The last four lines of the terminal output are enclosed in a red rectangular box.

Figure 6: Результат программы simpleid

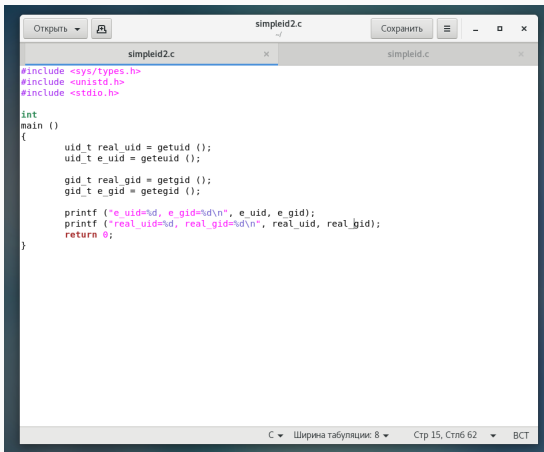
Работа с файлом simpleid2

A terminal window titled 'guest@ldabuzyarova:~' with a menu bar (Файл, Правка, Вид, Поиск, Терминал, Справка). The terminal shows the installation of the isl and cloog libraries, the installation of GCC 4.8.5, and the creation of a directory 'laba5'. Inside 'laba5', a file 'simpleid.c' is created and opened with gedit. The user then runs 'id', showing they are user '1001' in group '1001'. Finally, a file 'simpleid2.c' is created and opened with gedit. The last two commands are highlighted with a red rectangle.

```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
obj-x86_64-redhat-linux/isl-install --with-cloog=/builddir/build/BUILD/gcc-4.8.5-20150702/obj-x86_64-redhat-linux/cloog-install --enable-gnu-indirect-function --with-tune=generic --with-arch_32=x86-64 --build=x86_64-redhat-linux  
Модель многопоточности: posix  
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[guest@ldabuzyarova ~]$ getenforce  
Permissive  
[guest@ldabuzyarova ~]$ cd  
[guest@ldabuzyarova ~]$ mkdir laba5  
[guest@ldabuzyarova ~]$ touch simpleid.c  
[guest@ldabuzyarova ~]$ gedit simpleid.c  
  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ gcc simpleid.c -o simpleid  
[guest@ldabuzyarova ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@ldabuzyarova ~]$ id  
uid=1001(guest) gid=1001(guest) rгруппы=1001(guest) контекст=unconfined_u:unconfi  
ned_t:unconfined_c:50 50.c0.c1023  
[guest@ldabuzyarova ~]$ touch simpleid2.c  
[guest@ldabuzyarova ~]$ gedit simpleid2.c  
[guest@ldabuzyarova ~]$
```

Figure 7: Создание и редактирование файла simpleid2.c

Работа с файлом simpleid2



The image shows a code editor window with the title bar 'simpleid2.c'. The editor contains the following C code:

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

int
main ()
{
    uid_t real_uid = getuid ();
    uid_t e_uid = geteuid ();

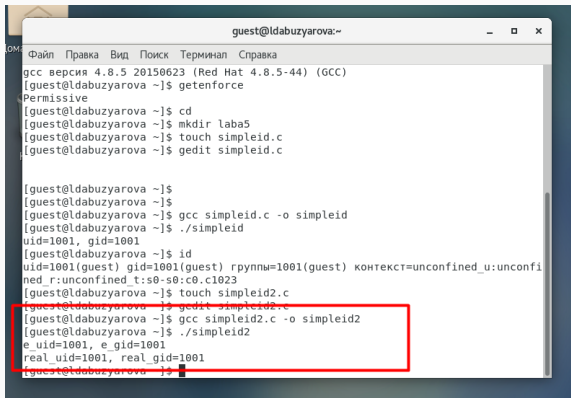
    gid_t real_gid = getgid ();
    gid_t e_gid = getegid ();

    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;
}
```

The editor interface includes a menu bar with 'Открыть' (Open) and 'Сохранить' (Save) buttons. The status bar at the bottom shows 'C', 'Ширина табуляции: 8', 'Стр 15, Стлб 62', and 'ВСТ'.

Figure 8: Написание программы

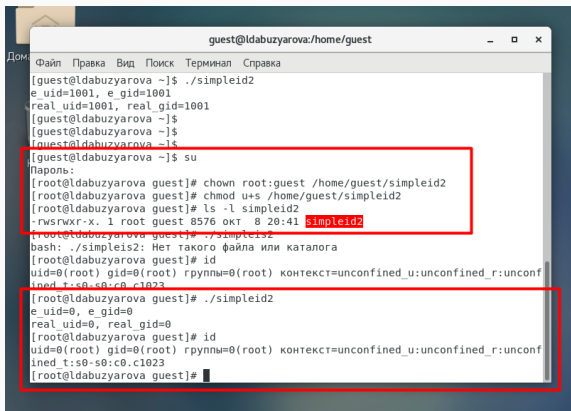
Работа с файлом simpleid2



```
guest@ldabuzyarova:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
gcc версия 4.8.5 20150623 (Red Hat 4.8.5-44) (GCC)  
[guest@ldabuzyarova ~]$ getenforce  
Permissive  
[guest@ldabuzyarova ~]$ cd  
[guest@ldabuzyarova ~]$ mkdir laba5  
[guest@ldabuzyarova ~]$ touch simpleid.c  
[guest@ldabuzyarova ~]$ gedit simpleid.c  
  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ gcc simpleid.c -o simpleid  
[guest@ldabuzyarova ~]$ ./simpleid  
uid=1001, gid=1001  
[guest@ldabuzyarova ~]$ id  
uid=1001(guest) gid=1001(guest) rпны=1001(guest) контекст=unconfined_u:unconfi  
ned_r:unconfined_t:s0-s0:c0.c1023  
[guest@ldabuzyarova ~]$ touch simpleid2.c  
[guest@ldabuzyarova ~]$ gedit simpleid2.c  
[guest@ldabuzyarova ~]$ gcc simpleid2.c -o simpleid2  
[guest@ldabuzyarova ~]$ ./simpleid2  
e_uid=1001, e_gid=1001  
real_uid=1001, real_gid=1001  
[guest@ldabuzyarova ~]$
```

Figure 9: Результат программы simpleid2 от guest

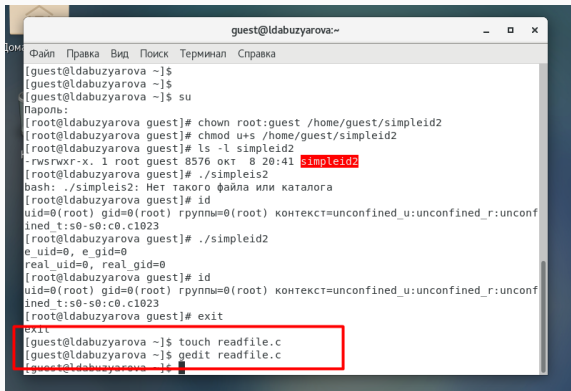
Работа с файлом simpleid2



```
guest@ldabuzyarova:~/home/guest
[guest@ldabuzyarova ~]$ ./simpleid2
e_uid=1001, e_gid=1001
real_uid=1001, real_gid=1001
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$ su
Пароль:
[root@ldabuzyarova guest]# chown root:guest /home/guest/simpleid2
[root@ldabuzyarova guest]# chmod u+s /home/guest/simpleid2
[root@ldabuzyarova guest]# ls -l simpleid2
-rwsrwxr-x. 1 root guest 8576 окт 8 20:41 simpleid2
[root@ldabuzyarova guest]# ./simpleid2
bash: ./simpleid2: Нет такого файла или каталога
[root@ldabuzyarova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ldabuzyarova guest]# ./simpleid2
e_uid=0, e_gid=0
real_uid=0, real_gid=0
[root@ldabuzyarova guest]# id
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@ldabuzyarova guest]#
```

Figure 10: Результат программы simpleid2 от root

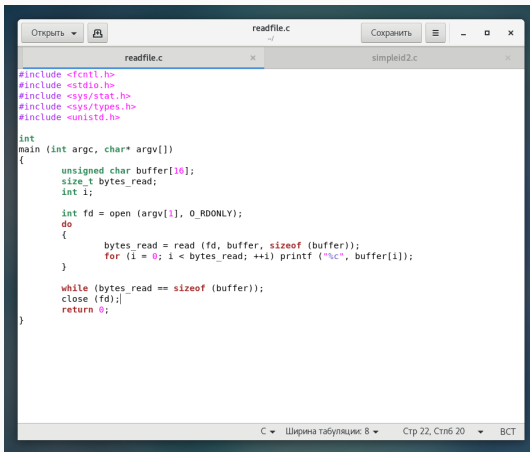
Работа с файлом readfile



```
guest@dabuzyarova:~  
Файл  Правка  Вид  Поиск  Терминал  Справка  
[guest@dabuzyarova ~]$  
[guest@dabuzyarova ~]$  
[guest@dabuzyarova ~]$ su  
Пароль:  
[root@dabuzyarova guest]# chown root:guest /home/guest/simpleid2  
[root@dabuzyarova guest]# chmod u+s /home/guest/simpleid2  
[root@dabuzyarova guest]# ls -l simpleid2  
-rwsrwxr-x. 1 root guest 8576 окт  8 20:41 simpleid2  
[root@dabuzyarova guest]# ./simpleis2  
bash: ./simpleis2: Нет такого файла или каталога  
[root@dabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@dabuzyarova guest]# ./simpleid2  
e_uid=0, e_gid=0  
real_uid=0, real_gid=0  
[root@dabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@dabuzyarova guest]# exit  
exit  
[guest@dabuzyarova ~]$ touch readfile.c  
[guest@dabuzyarova ~]$ gedit readfile.c  
[guest@dabuzyarova ~]$
```

Figure 11: Создание и редактирование файла readfile.c

Работа с файлом readfile

A screenshot of a code editor window titled 'readfile.c'. The editor has a menu bar with 'Открыть' (Open) and 'Сохранить' (Save) buttons. Below the menu bar are two tabs: 'readfile.c' (active) and 'simpleid2.c'. The code is written in C and uses syntax highlighting. It includes headers for `<fcntl.h>`, `<stdio.h>`, `<sys/stat.h>`, `<sys/types.h>`, and `<unistd.h>`. The `main` function takes `argc` and `argv` as arguments. It declares a `buffer` of type `unsigned char` with size 10, a `bytes_read` variable of type `size_t`, and an `i` variable of type `int`. It opens the file specified in `argv[1]` in read-only mode. A `do` loop reads data from the file into the buffer and prints each byte. A `while` loop continues until the end of the file is reached. The status bar at the bottom shows 'С', 'Ширина табуляции: 8', 'Стр 22, Стб 20', and 'ВСТ'.

```
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

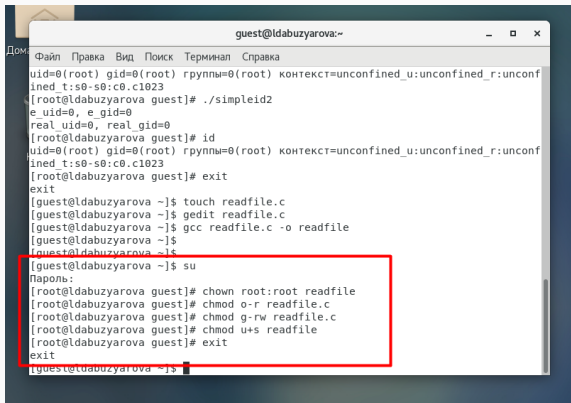
int
main (int argc, char* argv[])
{
    unsigned char buffer[10];
    size_t bytes_read;
    int i;

    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 12: Программа readfile

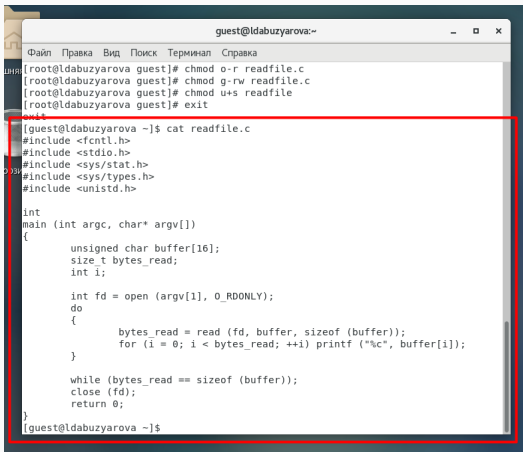
Работа с файлом readfile



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# ./simpleid2  
e_uid=0, e_gid=0  
real uid=0, real_gid=0  
[root@ldabuzyarova guest]# id  
uid=0(root) gid=0(root) группы=0(root) контекст=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ touch readfile.c  
[guest@ldabuzyarova ~]$ gedit readfile.c  
[guest@ldabuzyarova ~]$ gcc readfile.c -o readfile  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ su  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# chown root:root readfile  
[root@ldabuzyarova guest]# chmod o-r readfile.c  
[root@ldabuzyarova guest]# chmod g-rw readfile.c  
[root@ldabuzyarova guest]# chmod u+s readfile  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$
```

Figure 13: Смена прав на файле readfile

Работа с файлом readfile



The screenshot shows a terminal window titled 'guest@ldabuzyarova:~'. The terminal displays a series of commands and their outputs. First, the user runs 'chmod o-r readfile.c', 'chmod g-rw readfile.c', and 'chmod u+s readfile'. Then, the user runs 'exit'. Finally, the user runs 'cat readfile.c', which displays the contents of the file 'readfile.c'. The code is a C program that opens a file in read-only mode, reads it into a buffer, and prints the contents character by character. The code is as follows:

```
guest@ldabuzyarova:~$ cat readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

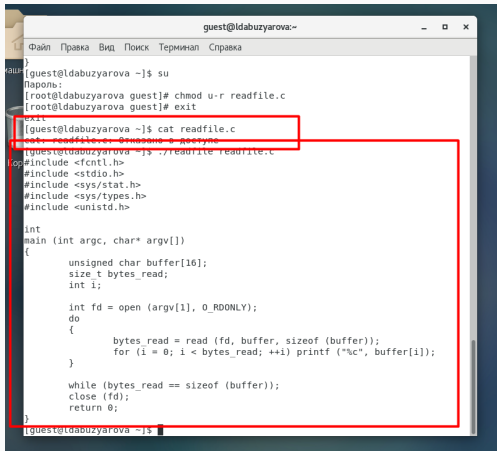
int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    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@ldabuzyarova:~$
```

Figure 14: Проверка файла на чтение

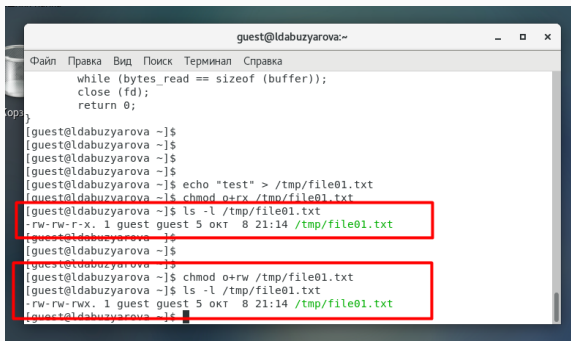
Работа с файлом readfile



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
[guest@ldabuzyarova ~]$ su  
Пароль:  
[root@ldabuzyarova guest]# chmod u-r readfile.c  
[root@ldabuzyarova guest]# exit  
exit  
[guest@ldabuzyarova ~]$ cat readfile.c  
cat: readfile.c: Permission denied  
[guest@ldabuzyarova ~]$ ./readfile readfile.c  
r.op#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;  
}
```

Figure 15: Повторная проверка файла на чтение

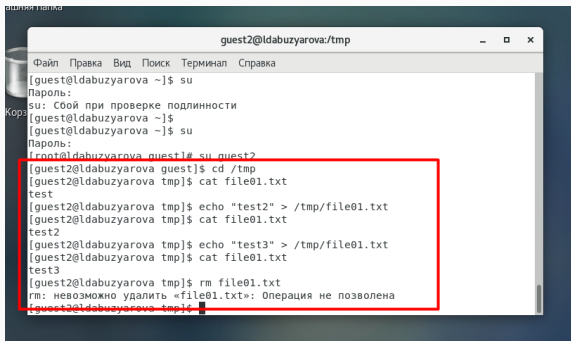
Исследование Sticky-бита



```
guest@ldabuzyarova:~  
Файл Правка Вид Поиск Терминал Справка  
while (bytes_read == sizeof (buffer));  
close (fd);  
return 0;  
}  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ echo "test" > /tmp/file01.txt  
[guest@ldabuzyarova ~]$ chmod o+rx /tmp/file01.txt  
[guest@ldabuzyarova ~]$ ls -l /tmp/file01.txt  
-rw-rw-r-x. 1 guest guest 5 окт  8 21:14 /tmp/file01.txt  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$  
[guest@ldabuzyarova ~]$ chmod o+rw /tmp/file01.txt  
[guest@ldabuzyarova ~]$ ls -l /tmp/file01.txt  
-rw-rw-rwx. 1 guest guest 5 окт  8 21:14 /tmp/file01.txt  
[guest@ldabuzyarova ~]$
```

Figure 16: Проверка прав на файле file01.txt

Исследование Sticky-бита

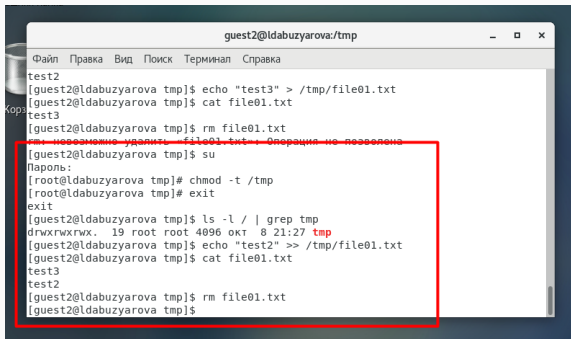


A terminal window titled 'guest2@ldabuzyarova:/tmp' showing a sequence of commands and their outputs. The window has a menu bar with 'Файл', 'Правка', 'Вид', 'Поиск', 'Терминал', and 'Справка'. The terminal session starts with a user switching from 'guest' to 'su' and then back to 'guest2'. A red rectangle highlights a series of commands where the user 'guest2' creates, writes to, and attempts to delete a file named 'file01.txt' in the '/tmp' directory. The file is created with 'echo', its contents are verified with 'cat', and an attempt to remove it with 'rm' results in a 'Permission denied' error, demonstrating the effect of the sticky bit.

```
guest2@ldabuzyarova: ~]$ su
Пароль:
su: Сбой при проверке подлинности
[guest@ldabuzyarova ~]$
[guest@ldabuzyarova ~]$ su
Пароль:
[root@ldabuzyarova guest]# su guest2
[guest2@ldabuzyarova guest]$ cd /tmp
[guest2@ldabuzyarova tmp]$ cat file01.txt
test
[guest2@ldabuzyarova tmp]$ echo "test2" > /tmp/file01.txt
[guest2@ldabuzyarova tmp]$ cat file01.txt
test2
[guest2@ldabuzyarova tmp]$ echo "test3" > /tmp/file01.txt
[guest2@ldabuzyarova tmp]$ cat file01.txt
test3
[guest2@ldabuzyarova tmp]$ rm file01.txt
rm: невозможно удалить «file01.txt»: Операция не позволена
[guest2@ldabuzyarova tmp]$
```

Figure 17: Дозапись и перезапись слов в файл file01.txt

Исследование Sticky-бита



```
guest2@ldabuzyarova:/tmp
Файл Правка Вид Поиск Терминал Справка
test2
[guest2@ldabuzyarova tmp]$ echo "test3" > /tmp/file01.txt
[guest2@ldabuzyarova tmp]$ cat file01.txt
test3
[guest2@ldabuzyarova tmp]$ rm file01.txt
rm: невозможно удалить 'file01.txt': Операция не разрешена
[guest2@ldabuzyarova tmp]$ su
Пароль:
[root@ldabuzyarova tmp]# chmod -t /tmp
[root@ldabuzyarova tmp]# exit
exit
[guest2@ldabuzyarova tmp]$ ls -l / | grep tmp
drwxrwxrwx. 19 root root 4096 окт 8 21:27 tmp
[guest2@ldabuzyarova tmp]$ echo "test2" >> /tmp/file01.txt
[guest2@ldabuzyarova tmp]$ cat file01.txt
test3
test2
[guest2@ldabuzyarova tmp]$ rm file01.txt
[guest2@ldabuzyarova tmp]$
```

Figure 18: Снятие атрибута t

Выводы

Изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Также мы рассмотрели работу механизма смены идентификатора процессов пользователей и влияние бита Sticky на запись и удаление файлов.