

Лабораторная работа №3

Дисциплина: Основы информационной безопасности

Феоктистов Владислав Сергеевич

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
3.1	Изменение атрибутов	7
3.2	Добавление пользователя в группу	8
3.3	Регистрация пользователя в новой группе	8
3.4	Таблицы	9
4	Выполнение лабораторной работы	12
4.1	Исполнение команд в консоли	12
4.2	Создание и использование скрипта	17
4.3	Таблицы прав и разрешенных действий	21
5	Выводы	29
	Список литературы	30

Список иллюстраций

4.1	Создание нового пользователя guest и пароля для него	12
4.2	Создание нового пользователя guest2 и пароля для него, а также добавление в группу	13
4.3	Информация о пользователях	14
4.4	Содержимое файла /etc/group	15
4.5	Регистрация пользователя в группе	16
4.6	Изменение прав директорий	16
4.7	Проверка существования файла и каталога	17
4.8	Копирование bash-скрипта	18
4.9	Создание bash-скриптов для автоматизации проверки	18
4.10	Создание bash-скриптов для автоматизации проверки	19
4.11	Создание bash-скриптов для автоматизации проверки	20
4.12	Запуск bash-скриптов	20

Список таблиц

3.1	Описание некоторых каталогов файловой системы GNU Linux . .	9
3.2	Описание некоторых используемых в работе команд	10
4.1	Установленные права и разрешенные действия для групп	21
4.2	Минимальные права для совершения операций от имени пользо- вателей входящих в группу	28

1 Цель работы

Целью данной работы является: приобретение практических навыков работы в консоли с правами и атрибутами файлов и каталогов для групп пользователей, закрепление теоретических основ дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux, проверка необходимых наборов прав для выполнения различных действий над файлами и каталогами для групп пользователей, получение навыков чтения выделенных прав через консоль.

2 Задание

Создать нового пользователя под именем guest2 с паролем и добавить его в группу guest; осуществить вход в систему от двух пользователей на двух различных консолях и получить информацию о них различными способами; выполнить регистрацию пользователя в группе; проверить и изменить права на существующие файлы и каталоги; заполнить таблицы разрешенных действий и минимальных прав для групп пользователей.

3 Теоретическое введение

3.1 Изменение атрибутов

В ОС Linux права доступа к файлам, атрибуты и владение управляют уровнем доступа, который система обрабатывает, а пользователи имеют к файлам. Это гарантирует, что только авторизованные пользователи и процессы могут получить доступ к определенным файлам и каталогам. Атрибуты состоят из девяти битов, которые и определяют права для разных групп пользователей. Первая тройка битов определяет права доступа для владельца, вторая тройка - для членов группы, последняя тройка - для всех остальных пользователей в системе. Каждая тройка битов (класс пользователей) определяет права на чтение, запись и исполнение. Эта концепция позволяет контролировать, какие пользователи могут читать, записывать (изменять) или выполнять файлы/каталоги.

Чтобы просмотреть права доступа к файлу, используется команда `ls` с опцией `-l`. Первый символ указывает тип файла. Это может быть обычный файл (`-`), каталог (`d`), символическая ссылка (`l`) или другие специфические типы файлов. Следующие девять символов предоставляют доступ к файлу, три тройки по три символа каждая (три пользователя, три типа прав: `r` - чтение, `w` - запись, `x` - исполнение).

Права доступа к файлу/каталогу можно изменить с помощью команды `chmod`. Только `root`, владелец файла или пользователь с привилегией `sudo` могут изменять права доступа к файлу или каталогу. Разрешения можно указывать с помощью символического, числового или справочного режимов [1].

3.2 Добавление пользователя в группу

Для каждого пользователя существует два типа групп - это первичная, основная для него группа, и вторичная, дополнительная.

- Первичная группа (основная) - создается автоматически, когда пользователь регистрируется в системе, в большинстве случаев имеет такое же имя, как и имя пользователя. Пользователь может иметь только одну основную группу;
- Вторичная группа - это дополнительные группы, к которым пользователь может быть добавлен в процессе работы.

Как обычно, лучше всего будет добавлять пользователя в группу через терминал, поскольку это даст вам больше гибкости и возможностей. Для изменения параметров пользователя используется команда `grpasswd` [2].

3.3 Регистрация пользователя в новой группе

Программа `newgrp` используется для изменения ID текущей группы в работающем сеансе. Если указан необязательный параметр `-`, то окружение пользователя будет инициализировано повторно, как если бы пользователь заново вошёл в систему, иначе имеющееся окружение, включая текущий рабочий каталог, изменено не будет.

Программа `newgrp` изменяет идентификатор текущей реальной группы на заданный или на группу по умолчанию, указанную в файле `/etc/passwd`, в случае если имя группы не указано. Программа `newgrp` также пытается добавить группу в список групп пользователя. Если пользователь не является суперпользователем, то его попросят ввести пароль, даже если он его не имеет (в файле `/etc/shadow`, если для этого пользователя имеется запись в файле теневых паролей, иначе используется файл `/etc/passwd`), а группа имеет, или если пользователь не является

членом группы, а группа имеет пароль. Если пользователь не является членом группы, а у группы пустой пароль, то пользователю будет отказано в доступе.

Если есть запись для этой группы в файле `/etc/gshadow`, то список членов и пароль этой группы будут взяты из этого файла, иначе используется запись из файла `/etc/group` [3].

3.4 Таблицы

Таблица 3.1: Описание некоторых каталогов файловой системы GNU Linux

Имя каталога	Описание каталога
<code>/</code>	Корневая директория, содержащая всю файловую систему
<code>/bin</code>	Основные системные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям
<code>/etc</code>	Общесистемные конфигурационные файлы и файлы конфигурации установленных программ
<code>/home</code>	Содержит домашние директории пользователей, которые, в свою очередь, содержат персональные настройки и данные пользователя
<code>/media</code>	Точки монтирования для сменных носителей
<code>/root</code>	Домашняя директория пользователя <code>root</code>
<code>/tmp</code>	Временные файлы
<code>/usr</code>	Вторичная иерархия для данных пользователя

Таблица 3.2: Описание некоторых используемых в работе команд

Ко- манда	Описание команды
useradd	Создание пользователя в Linux. Необходимо будет указать имя нового пользователя.
passwd	Создание и изменение пользовательских паролей. Необходимо будет указать имя пользователя, для которого нужно создать/изменить пароль.
groupadd	Добавление указанного в опции пользователя в указанную группу. Опция -G дополнительные группы для пользователя, -a добавляет пользователя в дополнительные группы из параметра -G, а не заменяет им текущее значение.
pwd	Выводит полный путь от корневого каталога к текущему рабочему каталогу: в контексте которого (по умолчанию) будут исполняться выводимые команды.
whoami	Отображает имя вошедшего в систему пользователя.
id	Выводит UID (идентификатор пользователя), GID (идентификатор группы пользователя), groups (основные группы пользователя)
groups	Выводит список групп, в которых состоит текущий пользователь или пользователь с указанным именем.
cat	Вывод содержимого указанного файла.
ls	Выводит содержимое каталога. Опция -l выводит дополнительную информацию, -a отображает скрытые файлы, в названии которых в самом начале стоит символ '.'
lsattr	Просмотр атрибутов файлов/каталогов в файловой системе Linux.
mkdir	Создание каталога по указанному пути и с указанным именем внутри пути.

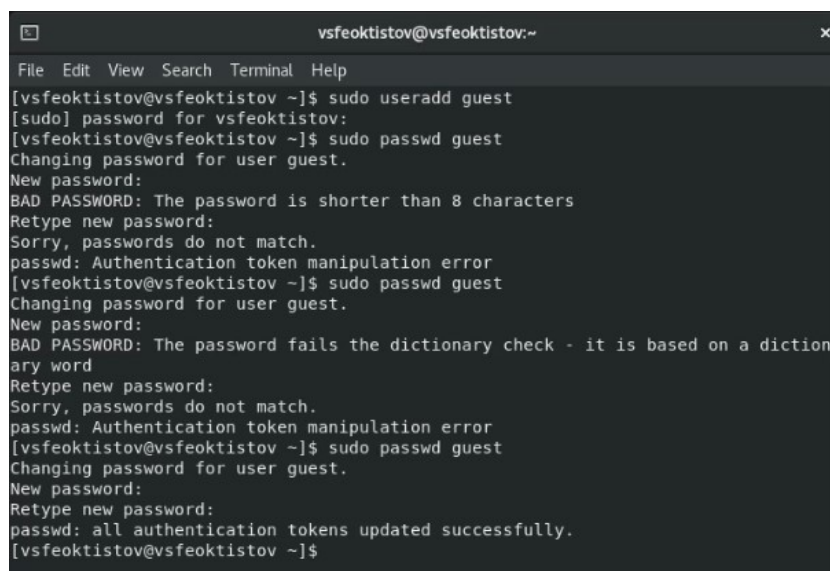
Ко-	
манда	Описание команды
<hr/>	
chmod	Изменение прав доступа к файлам и каталогам, используемых в Unix-подобных операционных системах.
echo	Вывод переданных аргументов, строки, текста.
chattr	Изменяет атрибуты файлов/каталогов в файловой системе Linux.
touch	Создает текстовый файл по указанному пути и с указанным именем внутри пути.
rm	Удаляет файл(ы) (каталог(и) при указании опции -r) по указанному(ым) пути(ям).
rename	Переименование файла/каталога. Формат rename [старое имя] [новое имя] [путь до файла].
cd	Перемещение по файловой системе.
grep	Дает возможность вести поиск строк. Также можно передать вывод любой команды в grep, что сильно упрощает работу во время поиска

Более подробно об Unix см. в [4–9].

4 Выполнение лабораторной работы

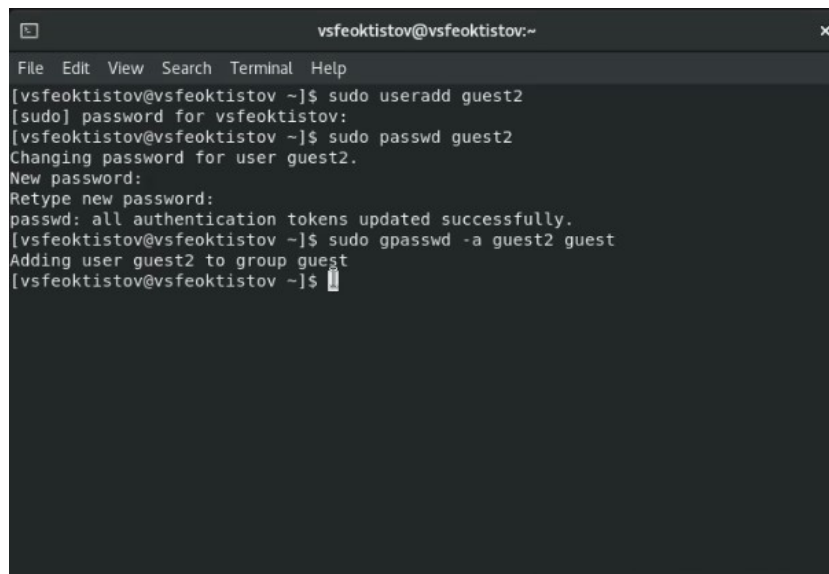
4.1 Исполнение команд в консоли

Так как в предыдущей лабораторной работе пользователь *guest* уже был создан, то создавать его по новой не нужно (рис. 4.1). Поэтому создаем только нового пользователя под именем *guest2* через команду `useradd` [cmd: `sudo useradd guest2`] и создаем для него пароль с помощью команды `passwd` [cmd: `sudo passwd guest2`]. После терминал попросит указать и подтвердить новый пароль. Затем добавляем пользователя *guest2* в группу *guest* командой `gpasswd` [cmd: `sudo passwd -a guest2 guest`] (рис. 4.2).



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ sudo useradd guest  
[sudo] password for vsfeoktistov:  
[vsfeoktistov@vsfeoktistov ~]$ sudo passwd guest  
Changing password for user guest.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
Sorry, passwords do not match.  
passwd: Authentication token manipulation error  
[vsfeoktistov@vsfeoktistov ~]$ sudo passwd guest  
Changing password for user guest.  
New password:  
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word  
Retype new password:  
Sorry, passwords do not match.  
passwd: Authentication token manipulation error  
[vsfeoktistov@vsfeoktistov ~]$ sudo passwd guest  
Changing password for user guest.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 4.1: Создание нового пользователя *guest* и пароля для него



```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ sudo useradd guest2  
[sudo] password for vsfeoktistov:  
[vsfeoktistov@vsfeoktistov ~]$ sudo passwd guest2  
Changing password for user guest2.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[vsfeoktistov@vsfeoktistov ~]$ sudo gpasswd -a guest2 guest  
Adding user guest2 to group guest  
[vsfeoktistov@vsfeoktistov ~]$
```

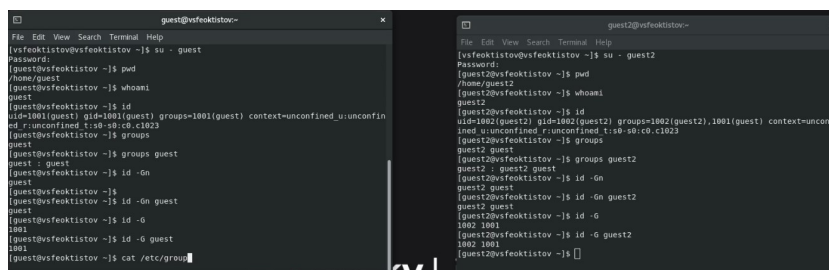
Рис. 4.2: Создание нового пользователя `guest2` и пароля для него, а также добавление в группу

Далее осуществляем вход в систему от двух пользователей на двух разных консолях: `guest` на первой консоли и `guest2` на второй. Сделать это можно, прописав команду `su - [имя пользователя]` [**cmd:** `su - guest` и `su - guest2`]. После чего получим некоторую информацию о этих пользователях (рис. 4.3):

- определим текущую директорию [**cmd:** `pwd`] (т.к. мы только что вошли в систему от имени другого пользователя, то, очевидно, что текущим каталогом будет домашний каталог текущего пользователя, т.е. для `guest` - `/home/guest`, для `guest2` - `/home/guest2`. Сравнивая вывод команды `pwd` с приглашение командной строки (набор символов перед знаком `$`, где имя перед знаком `@` - имя текущего пользователя, имя после `@` - имя хоста, после которого через пробел идет путь до текущего каталога), определяем, что текущий каталог из `pwd` совпадает с путем, указанным в приглашении (знак `~` - путь до домашнего каталога текущего пользователя));
- уточним имя пользователя [**cmd:** `whoami`];
- уточним группу пользователя [**cmd:** `id`]. Определяем это из значения переменной `gid`;

- уточним группы, в которые входит пользователь [**cmds:** *groups* или *groups [имя пользователя]*];

После чего сравним вывод команды *groups* с выводом команд *id -Gn* и *id -G* (или *id -Gn [имя пользователя]* и *id -G [имя пользователя]*). Из вывода команд видно, что команды *groups* и *id -Gn* одинаковые, т.е. выводят имена групп, в которых состоит пользователь, а команда *id -G* в отличие от них выводит *gid* групп.



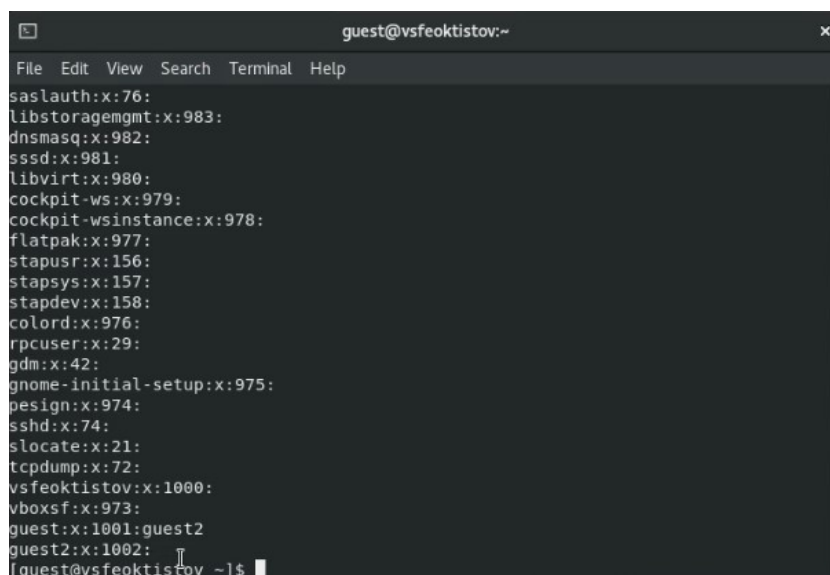
```

quest@vsfeaktistov:~$ su - quest
Password:
[quest@vsfeaktistov:~]$ pwd
/home/quest
[quest@vsfeaktistov:~]$ whoami
quest
[quest@vsfeaktistov:~]$ id
uid=1001(quest) gid=1001(quest) groups=1001(quest) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[quest@vsfeaktistov:~]$ groups
quest
[quest@vsfeaktistov:~]$ groups quest
quest : quest
[quest@vsfeaktistov:~]$ id -Gn
quest
[quest@vsfeaktistov:~]$ id -Gn quest
quest
[quest@vsfeaktistov:~]$ id -G
1001
[quest@vsfeaktistov:~]$ id -G quest
1001
[quest@vsfeaktistov:~]$ cat /etc/group
quest:x:1001:quest
quest2:x:1002:quest2

```

Рис. 4.3: Информация о пользователях

Полученную информацию сравним с содержимым файла */etc/group* [**cmd:** *cat /etc/group*]. Как видно, этот файл содержит информацию о всех группах в системе: их *gid* (Group identifier), а также какие пользователи состоят в этих группах. Т.е. команды *groups* и *id* выводят информацию о том, в каких группах состоит пользователь, в то время как файл */etc/group* содержит информацию о том, кто состоит в группах (рис. 4.4).

A terminal window titled 'guest@vsfeoktistov:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal displays the contents of the /etc/group file, listing system users and groups with their IDs and home directories. The list includes: saslauth:x:76:, libstorageengine:x:983:, dnsmasq:x:982:, sssd:x:981:, libvirt:x:980:, cockpit-ws:x:979:, cockpit-wsinstance:x:978:, flatpak:x:977:, stapusr:x:156:, stapys:x:157:, stapdev:x:158:, colord:x:976:, rpcuser:x:29:, gdm:x:42:, gnome-initial-setup:x:975:, pesign:x:974:, sshd:x:74:, slocate:x:21:, tcpdump:x:72:, vsfeoktistov:x:1000:, vboxsf:x:973:, guest:x:1001:guest2, and guest2:x:1002:. The prompt '[guest@vsfeoktistov ~]\$' is visible at the bottom.

```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
saslauth:x:76:  
libstorageengine:x:983:  
dnsmasq:x:982:  
sssd:x:981:  
libvirt:x:980:  
cockpit-ws:x:979:  
cockpit-wsinstance:x:978:  
flatpak:x:977:  
stapusr:x:156:  
stapys:x:157:  
stapdev:x:158:  
colord:x:976:  
rpcuser:x:29:  
gdm:x:42:  
gnome-initial-setup:x:975:  
pesign:x:974:  
sshd:x:74:  
slocate:x:21:  
tcpdump:x:72:  
vsfeoktistov:x:1000:  
vboxsf:x:973:  
guest:x:1001:guest2  
guest2:x:1002:  
[guest@vsfeoktistov ~]$
```

Рис. 4.4: Содержимое файла /etc/group

Далее переход во вторую консоль, в которой вошли в систему от имени пользователя *guest2*, и выполняем регистрацию пользователя *guest2* в группе *guest* командой `newgrp guest` (рис. 4.5). Из картинки 4.5 видно, что в результате меняется текущий идентификатор реальной группы на заданный (gid пользователя *guest2* поменялся с 1002 на 1001), но не меняется внутри файла */etc/passwd*. Если не указывать имя группы в команде `newgrp`, то установится идентификатор группы из файла */etc/passwd* для этого пользователя.

```
guest2@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest2@vsfeoktistov ~]$ id guest2  
uid=1002(guest2) gid=1002(guest2) groups=1002(guest2),1001(guest)  
[guest2@vsfeoktistov ~]$ cat /etc/passwd | -i "guest2"  
bash: -i: command not found...  
[guest2@vsfeoktistov ~]$ cat /etc/passwd | grep -i "guest2"  
guest2:x:1002:1002:~/home/guest2:/bin/bash  
[guest2@vsfeoktistov ~]$ newgrp guest  
[guest2@vsfeoktistov ~]$ id guest2  
uid=1002(guest2) gid=1002(guest2) groups=1002(guest2),1001(guest)  
[guest2@vsfeoktistov ~]$ cat /etc/passwd | grep -i "guest2"  
guest2:x:1002:1002:~/home/guest2:/bin/bash  
[guest2@vsfeoktistov ~]$ id  
uid=1002(guest2) gid=1001(guest) groups=1001(guest),1002(guest2) context=unconfi  
ned_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest2@vsfeoktistov ~]$ newgrp  
[guest2@vsfeoktistov ~]$ id  
uid=1002(guest2) gid=1002(guest2) groups=1002(guest2),1001(guest) context=unconf  
ined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[guest2@vsfeoktistov ~]$ newgrp guest  
[guest2@vsfeoktistov ~]$
```

Рис. 4.5: Регистрация пользователя в группе

От имени пользователя *guest* изменим права директории */home/guest*, разрешив все действия для пользователей группы [cmd: *chmod g+rwX /home/guest*], а также снимем с директории */home/guest/dir1* все атрибуты командой *chmod 000 dir1*. С помощью команд *ls -l* можно посмотреть как меняются атрибуты каталогов (рис. 4.6).

```
guest@vsfeoktistov:~  
File Edit View Search Terminal Help  
[guest@vsfeoktistov ~]$ chmod g+rwX /home/guest  
[guest@vsfeoktistov ~]$ pwd  
/home/guest  
[guest@vsfeoktistov ~]$ cd ../  
[guest@vsfeoktistov home]$ ls -l  
total 8  
drwxrwx---. 16 guest      guest      4096 Sep 21 22:41 guest  
drwx-----. 4 guest2     guest2     112 Sep 21 22:41 guest2  
drwx-----. 19 vsfeoktistov vsfeoktistov 4096 Sep 21 21:28 vsfeoktistov  
[guest@vsfeoktistov home]$ cd ~  
[guest@vsfeoktistov ~]$ ls  
Desktop Documents Music Public test.sh  
dir1 Downloads Pictures Templates Videos  
[guest@vsfeoktistov ~]$ chmod 000 dir1  
[guest@vsfeoktistov ~]$ ls -l  
total 4  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Desktop  
d-----.. 2 guest guest     19 Sep 17 13:41 dir1  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Documents  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Downloads  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Music  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Pictures  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Public  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Templates  
-rw-rw-r--. 1 guest guest    3103 Sep 17 14:05 test.sh  
drwxr-xr-x. 2 guest guest      6 Sep 17 12:13 Videos  
[guest@vsfeoktistov ~]$
```

Рис. 4.6: Изменение прав директорий

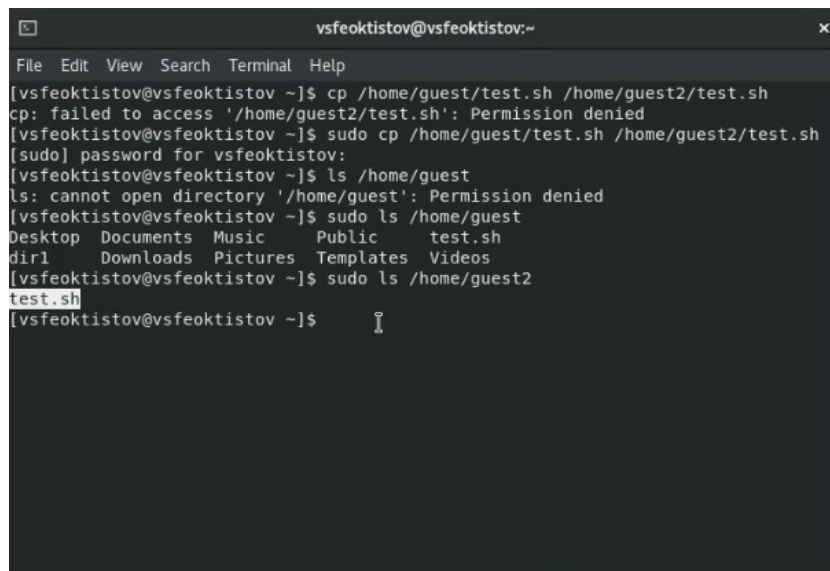
Директория /home/guest/dir1 и файл /home/guest/dir1/file1 были созданы еще в предыдущей лабораторной работе (рис. 4.7).

```
[guest@vsfeoktistov ~]$ ls -l dir1
total 4
----rwx---. 1 guest guest 13 Sep 18 14:02 file1
[guest@vsfeoktistov ~]$
```

Рис. 4.7: Проверка существования файла и каталога

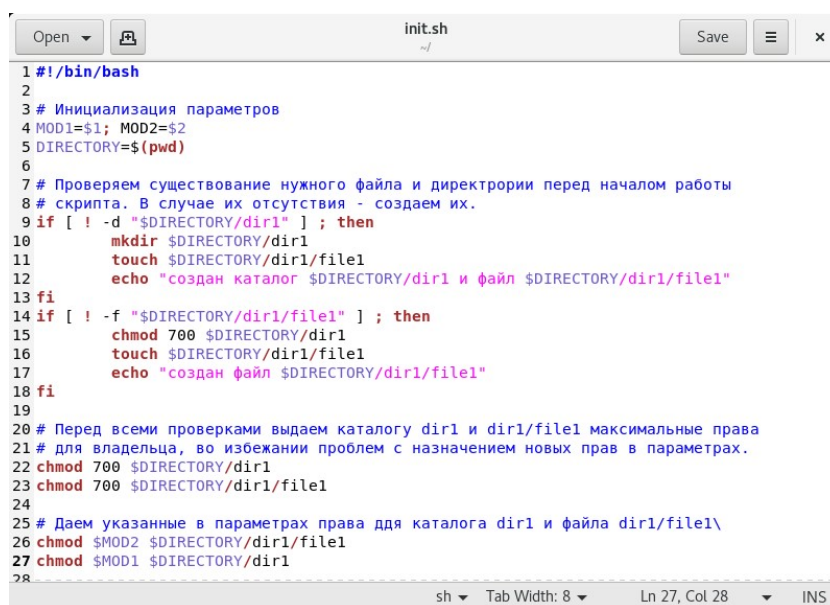
4.2 Создание и использование скрипта

Далее я изучил какие действия можно будет совершать над файлами/каталогами при различных комбинациях атрибутов прав доступа для групп. Для этого можно последовательно выполнить ряд команд: *touch* - попытка создать файл, *rm* - попытка удалить файл, *echo "" > /path* - попытка записать данные в файл, *cat* - попытка прочитать информацию из файла, *cd* - попытка перейти в директорию, *ls* - попытка просмотреть содержимое директории, *rename* - попытка переименовать файл, *chattr* - попытка изменить расширенные атрибуты файла. Но поскольку всего таких комбинаций атрибутов $88=64$, то учитывая то, что нужно еще заполнить 8 колонок, то понадобится исполнить не менее $888=512$ команд, что достаточно много. Поэтому я написал *bash* скрипт, который упрощает проверку. Причем подобный скрипт уже был написан в предыдущей лабораторной работе, поэтому будет достаточно только немного его отредактировать: скопировать его в домашний каталог пользователя *guest2*, убрать часть с инициализацией и добавить опцию указания пути, где будут проводиться проверки действий над файлом и каталогом, а в домашнем каталоге *guest** переименовать скрипт в *init.sh* и оставить только часть с инициализацией (рис. 4.8 - 4.11).



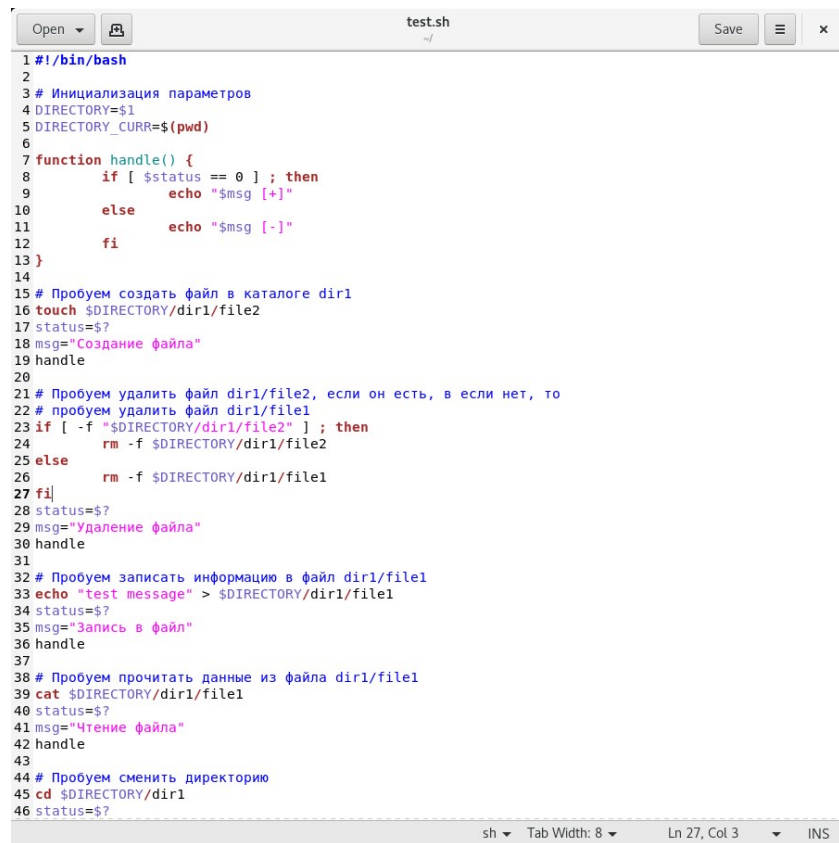
```
vsfeoktistov@vsfeoktistov:~  
File Edit View Search Terminal Help  
[vsfeoktistov@vsfeoktistov ~]$ cp /home/guest/test.sh /home/guest2/test.sh  
cp: failed to access '/home/guest2/test.sh': Permission denied  
[vsfeoktistov@vsfeoktistov ~]$ sudo cp /home/guest/test.sh /home/guest2/test.sh  
[sudo] password for vsfeoktistov:  
[vsfeoktistov@vsfeoktistov ~]$ ls /home/guest  
ls: cannot open directory '/home/guest': Permission denied  
[vsfeoktistov@vsfeoktistov ~]$ sudo ls /home/guest  
Desktop Documents Music Public test.sh  
dir1 Downloads Pictures Templates Videos  
[vsfeoktistov@vsfeoktistov ~]$ sudo ls /home/guest2  
test.sh  
[vsfeoktistov@vsfeoktistov ~]$
```

Рис. 4.8: Копирование bash-скрипта



```
init.sh  
~/  
Save  
x  
1 #!/bin/bash  
2  
3 # Инициализация параметров  
4 MOD1=$1; MOD2=$2  
5 DIRECTORY=$(pwd)  
6  
7 # Проверяем существование нужного файла и директории перед началом работы  
8 # скрипта. В случае их отсутствия - создаем их.  
9 if [ ! -d "$DIRECTORY/dir1" ] ; then  
10     mkdir $DIRECTORY/dir1  
11     touch $DIRECTORY/dir1/file1  
12     echo "создан каталог $DIRECTORY/dir1 и файл $DIRECTORY/dir1/file1"  
13 fi  
14 if [ ! -f "$DIRECTORY/dir1/file1" ] ; then  
15     chmod 700 $DIRECTORY/dir1  
16     touch $DIRECTORY/dir1/file1  
17     echo "создан файл $DIRECTORY/dir1/file1"  
18 fi  
19  
20 # Перед всеми проверками выдаем каталогу dir1 и dir1/file1 максимальные права  
21 # для владельца, во избежании проблем с назначением новых прав в параметрах.  
22 chmod 700 $DIRECTORY/dir1  
23 chmod 700 $DIRECTORY/dir1/file1  
24  
25 # Даем указанные в параметрах права для каталога dir1 и файла dir1/file1\  
26 chmod $MOD2 $DIRECTORY/dir1/file1  
27 chmod $MOD1 $DIRECTORY/dir1  
28  
sh Tab Width: 8 Ln 27, Col 28 INS
```

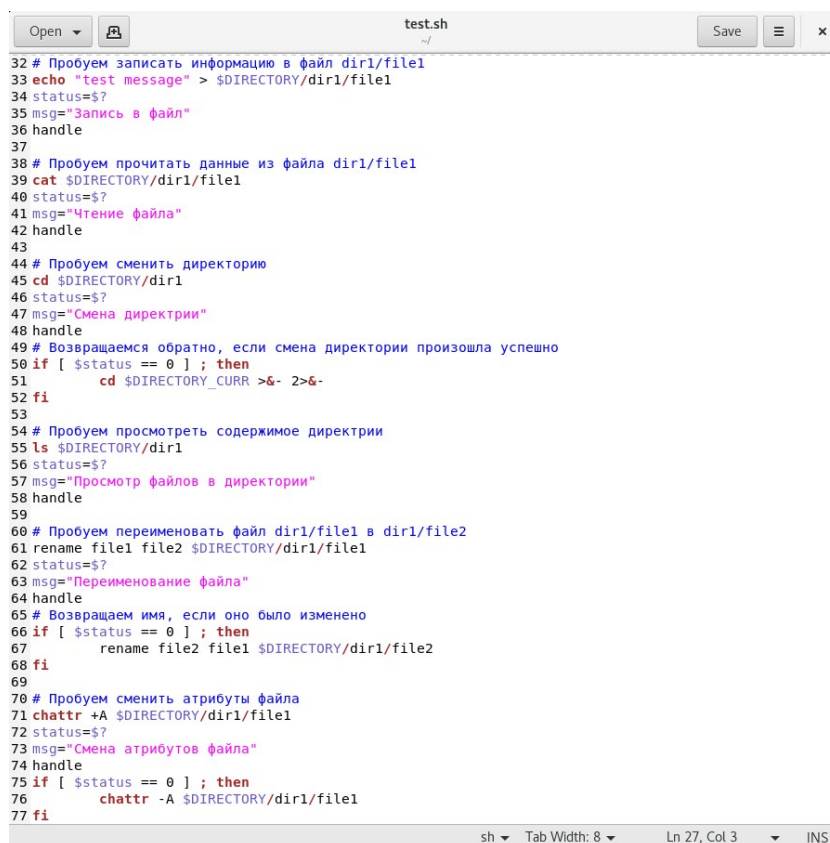
Рис. 4.9: Создание bash-скриптов для автоматизации проверки



```
1 #!/bin/bash
2
3 # Инициализация параметров
4 DIRECTORY=$1
5 DIRECTORY_CURR=$(pwd)
6
7 function handle() {
8     if [ $status == 0 ] ; then
9         echo "$msg [+]"
10    else
11        echo "$msg [-]"
12    fi
13 }
14
15 # Пробуем создать файл в каталоге dir1
16 touch $DIRECTORY/dir1/file2
17 status=$?
18 msg="Создание файла"
19 handle
20
21 # Пробуем удалить файл dir1/file2, если он есть, в если нет, то
22 # пробуем удалить файл dir1/file1
23 if [ -f "$DIRECTORY/dir1/file2" ] ; then
24     rm -f $DIRECTORY/dir1/file2
25 else
26     rm -f $DIRECTORY/dir1/file1
27 fi
28 status=$?
29 msg="Удаление файла"
30 handle
31
32 # Пробуем записать информацию в файл dir1/file1
33 echo "test message" > $DIRECTORY/dir1/file1
34 status=$?
35 msg="Запись в файл"
36 handle
37
38 # Пробуем прочитать данные из файла dir1/file1
39 cat $DIRECTORY/dir1/file1
40 status=$?
41 msg="Чтение файла"
42 handle
43
44 # Пробуем сменить директорию
45 cd $DIRECTORY/dir1
46 status=$?
```

sh Tab Width: 8 Ln 27, Col 3 INS

Рис. 4.10: Создание bash-скриптов для автоматизации проверки



```
32 # Пробуем записать информацию в файл dir1/file1
33 echo "test message" > $DIRECTORY/dir1/file1
34 status=$?
35 msg="Запись в файл"
36 handle
37
38 # Пробуем прочитать данные из файла dir1/file1
39 cat $DIRECTORY/dir1/file1
40 status=$?
41 msg="Чтение файла"
42 handle
43
44 # Пробуем сменить директорию
45 cd $DIRECTORY/dir1
46 status=$?
47 msg="Смена директрии"
48 handle
49 # Возвращаемся обратно, если смена директории произошла успешно
50 if [ $status == 0 ] ; then
51     cd $DIRECTORY_CURR >&- 2>&-
52 fi
53
54 # Пробуем просмотреть содержимое директрии
55 ls $DIRECTORY/dir1
56 status=$?
57 msg="Просмотр файлов в директрии"
58 handle
59
60 # Пробуем переименовать файл dir1/file1 в dir1/file2
61 rename file1 file2 $DIRECTORY/dir1/file1
62 status=$?
63 msg="Переименование файла"
64 handle
65 # Возвращаем имя, если оно было изменено
66 if [ $status == 0 ] ; then
67     rename file2 file1 $DIRECTORY/dir1/file2
68 fi
69
70 # Пробуем сменить атрибуты файла
71 chattr +A $DIRECTORY/dir1/file1
72 status=$?
73 msg="Смена атрибутов файла"
74 handle
75 if [ $status == 0 ] ; then
76     chattr -A $DIRECTORY/dir1/file1
77 fi
```

Рис. 4.11: Создание bash-скриптов для автоматизации проверки

Запустить эти скрипты можно с помощью команд [**cmds:** `sh init.sh 000 000` в первом терминале и `sh test.sh /home/guest` во втором терминале]. Таким образом, скрипт `init.sh` снимет все права для файла `/home/guest/dir1/file1` и каталога `/home/guest/dir1`, а скрипт `test.sh` показывает какие действия можно будет над ними выполнять (рис. 4.12).

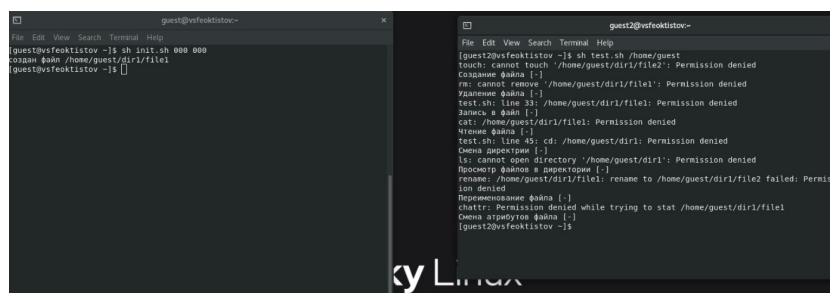


Рис. 4.12: Запуск bash-скриптов

4.3 Таблицы прав и разрешенных действий

Таблица 4.1: Установленные права и разрешенные действия для групп

Пра- ва		Со- зда- ние	Уда- ле- ние	За- пись в	Чте- ние	Сме- на	Про- смотр	Пере- имено- вание	Смена атрибу- тов
ди- рек- то- рии	Права файла	фай- ла	фай- ла	файл	фай- ла	рек- то- рии	файлов в директо- рии	файла	файла
d---	----	-	-	-	-	-	-	-	-
(000)	(000)								
d--x-	----	-	-	-	-	+	-	-	-
(010)	(000)								
d--	----	-	-	-	-	-	-	-	-
w--	(000)								
(020)									
d--	----	+	+	-	-	+	-	+	-
wx-	(000)								
(030)									
d-r--	----	-	-	-	-	-	+	-	-
(040)	(000)								
d-r-	----	-	-	-	-	+	+	-	-
x-	(000)								
(050)									
d-rw-	----	-	-	-	-	-	+	-	-
(060)	(000)								
d-rwx-	----	+	+	-	-	+	+	+	-
(070)	(000)								

Пра-						Сме-			
ва		Со-	Уда-			на	Про-		
ди-		зда-	ле-	За-	Чте-	ди-	смотр	Пере-	Смена
рек-		ние	ние	пись	ние	рек-	файлов в	имено-	атрибу-
то-	Права	фай-	фай-	в	фай-	то-	директо-	вание	тов
рии	файла	ла	ла	файл	ла	рии	рии	файла	файла

d---	--x-	-	-	-	-	-	-	-	-
(000)	(010)								
d--x-	--x-	-	-	-	-	+	-	-	-
(010)	(010)								
d--	--x-	-	-	-	-	-	-	-	-
w--	(010)								
(020)									
d--	--x-	+	+	-	-	+	-	+	-
wx-	(010)								
(030)									
d-r--	--x-	-	-	-	-	-	+	-	-
(040)	(010)								
d-r-	--x-	-	-	-	-	+	+	-	-
x-	(010)								
(050)									
d-rw---	--x-	-	-	-	-	-	+	-	-
(060)	(010)								
d-rwx---	--x-	+	+	-	-	+	+	+	-
(070)	(010)								

d---	--w--	-	-	-	-	-	-	-	-
(000)	(020)								

Пра- ва		Со- зда- ние	Уда- ле- ние	За- пись	Чте- ние	Сме- на	Про- смотр	Пере- имено- вание	Смена атрибу- тов
ди- рек- то- рии	Права файла	фай- ла	фай- ла	в файл	фай- ла	ди- рек- то- рии	файлов в директо- рии	файла	файла
d--x--w--	(010)	-	-	+	-	+	-	-	-
d--w--	(020)	-	-	-	-	-	-	-	-
d--w--	(020)	+	+	+	-	+	-	+	-
wx--	(020)	-	-	-	-	-	+	-	-
d-r--w--	(040)	-	-	+	-	+	+	-	-
d-r-x--	(050)	-	-	-	-	-	+	-	-
d-rw--w--	(060)	+	+	+	-	+	+	+	-
d-rwx--w--	(070)	-	-	-	-	-	-	-	-
d--x--wx--	(010)	-	-	+	-	+	-	-	-
d--x--wx--	(030)	-	-	+	-	+	-	-	-

Права		Смешанные				Смена			
ди-рек-то-рии	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	ди-рек-то-рии	Просмотр файлов в директории	Переименование файла	Смена атрибутов файла
d--w--(020)	--wx--(030)	-	-	-	-	-	-	-	-
d--wx--(030)	--wx--(030)	+	+	+	-	+	-	+	-
d-r--(040)	--wx--(030)	-	-	-	-	-	+	-	-
d-r-x--(050)	--wx--(030)	-	-	+	-	+	+	-	-
d-rw--(060)	--wx--(030)	-	-	-	-	-	+	-	-
d-rwx--(070)	--wx--(030)	+	+	+	-	+	+	+	-

d----(000)	--r--(040)	-	-	-	-	-	-	-	-
d--x--(010)	--r--(040)	-	-	-	+	+	-	-	-
d--w--(020)	--r--(040)	-	-	-	-	-	-	-	-

Права		Смешанные				Смена			
ди-рек-то-рии	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	ди-рек-то-рии	Про-смотр файлов в директо-рии	Пере-имено-вание файла	Смена атрибу-тов файла
d--wx--	--r-- (040)	+	+	-	+	+	-	+	-
d-r--	--r-- (040)	-	-	-	-	-	+	-	-
d-r-x--	--r-- (040)	-	-	-	+	+	+	-	-
d-rw--	--r-- (040)	-	-	-	-	-	+	-	-
d-rwx--	--r-- (040)	+	+	-	+	+	+	+	-
<hr/>									
d---wx--	--r-x-- (050)	-	-	-	-	-	-	-	-
d---wx--	--r-x-- (050)	-	-	-	+	+	-	-	-
d---w---	--r-x-- (050)	-	-	-	-	-	-	-	-
d---wx--	--r-x-- (050)	+	+	-	+	+	-	+	-

Права		Смешанные				Смена			
ди-рек-то-рии	Права файла	Создание файла	Удаление файла	Запись в файл	Чтение файла	ди-рек-то-рии	Про-смотр файлов в директо-рии	Пере-имено-вание файла	Смена атрибу-тов файла
d-r--	--r-x-	-	-	-	-	-	+	-	-
(040)	(050)								
d-r-x-	--r-x-	-	-	-	+	+	+	-	-
x-	(050)								
(050)									
d-rw--	--r-x-	-	-	-	-	-	+	-	-
(060)	(050)								
d-rwx-	--r-x-	+	+	-	+	+	+	+	-
(070)	(050)								

d----	--rw--	-	-	-	-	-	-	-	-
(000)	(060)								
d--x-	--rw--	-	-	+	+	+	-	-	-
(010)	(060)								
d--	--rw--	-	-	-	-	-	-	-	-
w--	(060)								
(020)									
d--	--rw--	+	+	+	+	+	-	+	-
wx-	(060)								
(030)									
d-r--	--rw--	-	-	-	-	-	+	-	-
(040)	(060)								

Права		Создание		Удаление		Запись		Чтение		Смена		Про- смотр		Переименование		Смена атрибутов	
ди-рек-то-рии	Права файла	фай-ла	фай-ла	в файл	фай-ла	то-рии	фай-ла	то-рии	фай-ла	то-рии	фай-ла	директо-рии	файла	имено-вание файла	файла	атрибу-тов файла	файла
d-r-x-	-rw-- (060)	-	-	+	+	+	+	+	+	+	+	+	+	-	-	-	-
d-rw-	-rw-- (060)	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-
d-rwx-	-rw-- (070)	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-
d---	-rwx-- (000)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d--x-	-rwx-- (010)	-	-	+	+	+	+	+	+	+	+	-	-	-	-	-	-
d--w-	-rwx-- (020)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d--wx-	-rwx-- (030)	+	+	+	+	+	+	+	+	+	+	-	-	+	+	-	-
d-r--	-rwx-- (040)	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-
d-r-x-	-rwx-- (050)	-	-	+	+	+	+	+	+	+	+	+	+	-	-	-	-

Права		Создание	Удаление	Запись	Чтение	Смена	Про- смотр	Переименование	Смена атрибутов
ди-рек-то-рии	Права файла	файла	файла	в файл	файла	ди-рек-то-рии	файлов в директо-рии	файла	файла
d—rw—	—rwx—	-	-	-	-	-	+	-	-
(060)	(070)								
d—rwx—	—rwx—	+	+	+	+	+	+	+	-
(070)	(070)								

Таблица 4.2: Минимальные права для совершения операций от имени пользователей входящих в группу

Операция	Минимальные права на директорию	Минимальные права на файл
Создание файла	d -wx (030)	— (000)
Удаление файла	d -wx (030)	— (000)
Чтение файла	d -x (010)	r— (040)
Запись в файл	d -x (010)	-w- (020)
Переименование файла	d -wx (030)	— (000)
Создание поддиректории	d -wx (030)	— (000)
Удаление поддиректории	d -wx (030)	— (000)

5 Выводы

В процессе выполнения лабораторной работы я приобрел практические навыки работы в консоли с правами и атрибутами файлов и каталогов для групп пользователей, закрепил теоретические основы дискреционного разграничения доступа в современных системах с открытым кодом на базе ОС Linux, проверил необходимый набор прав для выполнения различных действий над файлами и каталогами, получил навыки чтения выделенных прав через консоль.

Список литературы

1. Понимание прав доступа к файлам в Linux [Электронный ресурс]. Baks, 2021. URL: <https://baks.dev/article/terminal/understanding-linux-file-permissions?ysclid=l8czjs1hnp553393513>.
2. Как добавить пользователя в группу Linux [Электронный ресурс]. Losst, 2017. URL: <https://losst.ru/kak-dobavit-polzovatelya-v-gruppu-linux?ysclid=l8czmv7y3f708653349>.
3. Программа newgrp [Электронный ресурс]. Ubuntu Manpage, 2019. URL: <https://manpages.ubuntu.com/manpages/bionic/ru/man1/newgrp.1.html>.
4. GNU Bash Manual [Электронный ресурс]. Free Software Foundation, 2016. URL: <https://www.gnu.org/software/bash/manual/>.
5. Newham C. Learning the bash Shell: Unix Shell Programming. O'Reilly Media, 2005. 354 с.
6. Zarrelli G. Mastering Bash. Packt Publishing, 2017. 502 с.
7. Robbins A. Bash Pocket Reference. O'Reilly Media, 2016. 156 с.
8. Таненбаум Э. Архитектура компьютера. 6-е изд. СПб.: Питер, 2013. 874 с.
9. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. СПб.: Питер, 2015. 1120 с.