

Отчёта по лабораторной работе 3

1132232887

Накова Амина Михайловна

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12
	Список литературы	13

Список иллюстраций

2.1 шаг 1	7
2.2 шаг 2	7
2.3 шаг 3	8
2.4 шаг 4	8
2.5 шаг 5	8
2.6 шаг 6	9
2.7 шаг 7	9
2.8 шаг 8	10

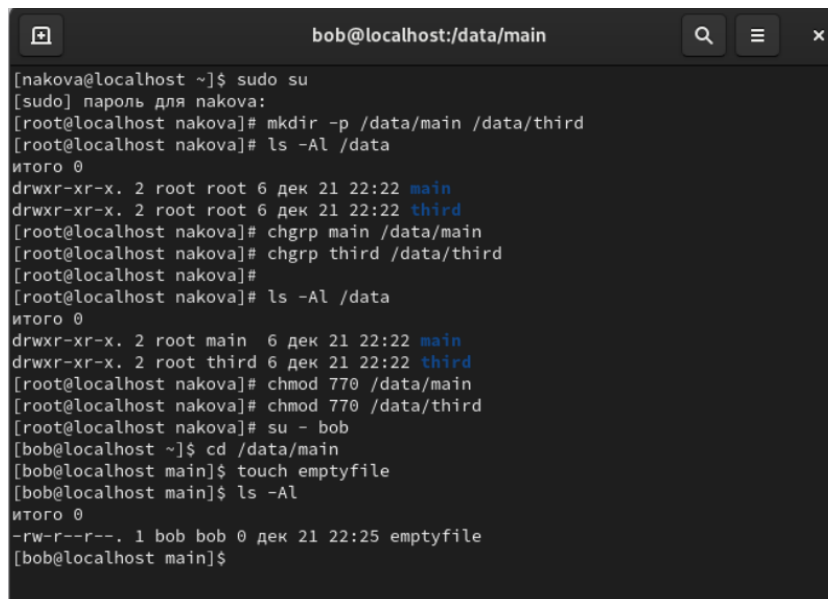
Список таблиц

1 Цель работы

Целью данной работы является получение навыков настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux.

2 Выполнение лабораторной работы

Открываем терминал с учётной записью root: `su -`. В корневом каталоге создаём каталоги `/data/main` и `/data/third` командой: `mkdir -p /data/main /data/third`. Посмотрим, кто является владельцем этих каталогов. Для этого используем: `ls -Al /data`. Владелец каталогов является суперпользователь. Прежде чем устанавливать разрешения, изменим владельцев этих каталогов с root на main и third соответственно: `chgrp main /data/main` и `chgrp third /data/third`. Теперь владельцем этих каталогов является main и third. Далее установим разрешения, позволяющие владельцам каталогов записывать файлы в эти каталоги и запрещающие доступ к содержимому каталогов всем другим пользователям и группам: `chmod 770 /data/main` и `chmod 770 /data/third`. Проверим установленные права доступа (рис. 2.1):

A terminal window titled 'bob@localhost:/data/main' showing a series of commands and their outputs. The user 'nakova' runs 'sudo su' to become root. Root creates the directory '/data/third' and sets group ownership for both '/data/main' and '/data/third' to 'main'. Root then changes permissions of both directories to '770' and switches back to the user 'bob'. Bob changes to the '/data/main' directory, creates an 'emptyfile', and lists the directory contents, showing only 'emptyfile' with permissions '-rw-r--r--' owned by 'bob' and group 'main'.

```
bob@localhost:/data/main
[nakova@localhost ~]$ sudo su
[sudo] пароль для nakova:
[root@localhost nakova]# mkdir -p /data/main /data/third
[root@localhost nakova]# ls -Al /data
итого 0
drwxr-xr-x. 2 root root 6 дек 21 22:22 main
drwxr-xr-x. 2 root root 6 дек 21 22:22 third
[root@localhost nakova]# chgrp main /data/main
[root@localhost nakova]# chgrp third /data/third
[root@localhost nakova]#
[root@localhost nakova]# ls -Al /data
итого 0
drwxr-xr-x. 2 root main 6 дек 21 22:22 main
drwxr-xr-x. 2 root third 6 дек 21 22:22 third
[root@localhost nakova]# chmod 770 /data/main
[root@localhost nakova]# chmod 770 /data/third
[root@localhost nakova]# su - bob
[bob@localhost ~]$ cd /data/main
[bob@localhost main]$ touch emptyfile
[bob@localhost main]$ ls -Al
итого 0
-rw-r--r--. 1 bob bob 0 дек 21 22:25 emptyfile
[bob@localhost main]$
```

Рис. 2.1: шаг 1

В другом терминале перейдём под учётную запись пользователя bob: `su - bob`. Под пользователем bob попробуем перейти в каталог `/data/main` и создать файл `emptyfile` в этом каталоге: `cd /data/main` и `touch emptyfile`. Так как пользователь bob является владельцем каталога `main`, нам удалось перейти в этот каталог и создать в нём новый файл. Теперь под пользователем bob попробуем перейти в каталог `/data/third` и создать файл `emptyfile` в этом каталоге. Так как пользователь bob не является владельцем каталога `third`, нам не удалось перейти в этот каталог и создать в нём новый файл (рис. 2.2):

A terminal window showing the continuation of the previous steps. The user 'bob' runs 'touch emptyfile' and 'ls' in the '/data/main' directory. Then, the user runs 'cd /data/third', which results in a 'Permission denied' error.

```
touch emptyfile
ls

cd /data/third
: Permission denied
```

Рис. 2.2: шаг 2

В другом терминале, под учётной записью пользователя bob (пользователь bob является членом группы `main`, как и `alice`) перейдём в каталог `/data/main`: `cd /data/main` (данный каталог уже был открыт в нашем терминале) и в этом

каталоге введём: `ls`. Мы увидим два файла, созданные пользователем `alice`. Теперь попробуем удалить файлы, принадлежащие пользователю `alice` командой: `rm -f alice*`. Убедимся, что файлы будут удалены пользователем `bob`. После проверки командой `ls` создадим два файла, которые принадлежат пользователю `bob`: `touch bob1` и `touch bob2` (рис. 2.3):

```
ls
ile
rm -f alice*
ls

touch bob1
touch bob2
```

Рис. 2.3: шаг 3

Переключаемся на учётную запись пользователя `alice` командой: `su alice`. Создаём пользователя `bob`: `sudo useradd bob`. При запросе вводим пароль пользователя. Проверяем, что пользователь `bob` создан (`id bob`) и устанавливаем пароль для пользователя: `sudo passwd bob` (рис. 2.4):

```
chmod g+s,o+t /data/main
```

Рис. 2.4: шаг 4

В терминале под пользователем `root` установим для каталога `/data/main` бит идентификатор группы, а также `sticky`-бит для разделяемого (общего) каталога группы: `chmod g+s,o+t /data/main` (рис. 2.5):

```
[nakova@localhost ~]$ touch alice3
[nakova@localhost ~]$ touch alice4
[nakova@localhost ~]$ ls
alice3  Видео      Загрузки    Музыка      'Рабочий стол'
alice4  Документы  Изображения Общедоступные Шаблоны
[nakova@localhost ~]$ la -Al /data
bash: la: команда не найдена...
[nakova@localhost ~]$ rm -rf bob*
[nakova@localhost ~]$
```

Рис. 2.5: шаг 5

Переходим в терминал под пользователем alice и создаём в каталоге /data/main файлы alice3 и alice4: touch alice3 и touch alice4. Теперь мы должны увидеть, что два созданных вами файла принадлежат группе main, которая является группой-владельцем каталога /data/main: ls и ls -Al /data. В этом же терминале попробуем удалить файлы, принадлежащие пользователю bob: rm -rf bob*. Убедимся, что sticky-bit предотвратит удаление этих файлов пользователем alice, поскольку этот пользователь не является владельцем этих файлов (Operation not permitted) (рис. 2.6):

```
# setfacl -m g:third:rx /data/main
# setfacl -m g:main:rx /data/third
```

Рис. 2.6: шаг 6

Откроем терминал с учётной записью root и установим права на чтение и выполнение в каталоге /data/main для группы third и права на чтение и выполнение для группы main в каталоге /data/third: setfacl -m g:third:rx /data/main и setfacl -m g:main:rx /data/third (Рис. 7.1). Теперь используем команду getfacl, чтобы убедиться в правильности установки разрешений: getfacl /data/main и getfacl /data/third (рис. 2.7):

```
getfacl: Removing leading '/' from absolute path names
# file: data/main
# owner: root
# group: main
# flags: -st
user::rwx
group::rwx
group:third:r-x
mask::rwx
other:---
```

Рис. 2.7: шаг 7

Установим ACL по умолчанию для каталога /data/main: setfacl -m d:g:third:rwx /data/main и для каталога /data/third: setfacl -m d:g:main:rwx /data/third. Убедимся, что настройки ACL работают, добавив новый файл в каталог /data/main: touch /data/main/newfile2. Используем getfacl /data/main/newfile2 (Рис. 9.1) для проверки текущих назначений полномочий. Выполним аналогичные действия для каталога /data/third (рис. 2.8):

```
getfacl: Removing leading '/' from absolute path names
# file: data/main/newfile1
# owner: root
# group: main
user::rw-
group::r--
other::r--
```

Рис. 2.8: шаг 8

Ответы на контрольные вопросы: 1. Как следует использовать команду `chown`, чтобы установить владельца группы для файла? Приведите пример. `chown bob:main /data/third/newfile`. 2. С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример. `find ~ -user bob -print`. 3. Как применить разрешения на чтение, запись и выполнение для всех файлов в каталоге `/data` для пользователей и владельцев групп, не устанавливая 9 никаких прав для других? Приведите пример. `chmod 770 /data` (скриншот из лабораторной работы). 4. Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым? `chmod +x file`. 5. Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример. `getfacl "имя каталога"`. 6. Необходимо, чтобы пользователи могли удалять только те файлы, владельцами которых они являются, или которые находятся в каталоге, владельцами которого они являются. С помощью какой команды можно это сделать? Приведите пример. `chmod g+s,o+t /data/main`. 7. Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге? `setfacl -m g:group:r .`. 8. Что нужно сделать для гарантии того, что члены группы получают разрешения на чтение для всех файлов в текущем каталоге и во всех его подкаталогах, а также для всех файлов, которые будут созданы в этом каталоге в будущем? Приведите пример. `setfacl -dm g:group:r /dir`. 9. Какое значение `umask` нужно установить, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы? Приведите пример. `007`. 10. Какая команда гарантирует, что никто не сможет удалить файл `myfile` случайно?

sudo chattr +i myfile.

3 Выводы

В ходе выполнения лабораторной работы были получены навыки настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux

Список литературы