

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

дисциплина: Основы администрирования операционных систем

Студент: Накова Амина Михайловна

Студ. билет № 1132232887

Группа: НПИбд-02-23

МОСКВА

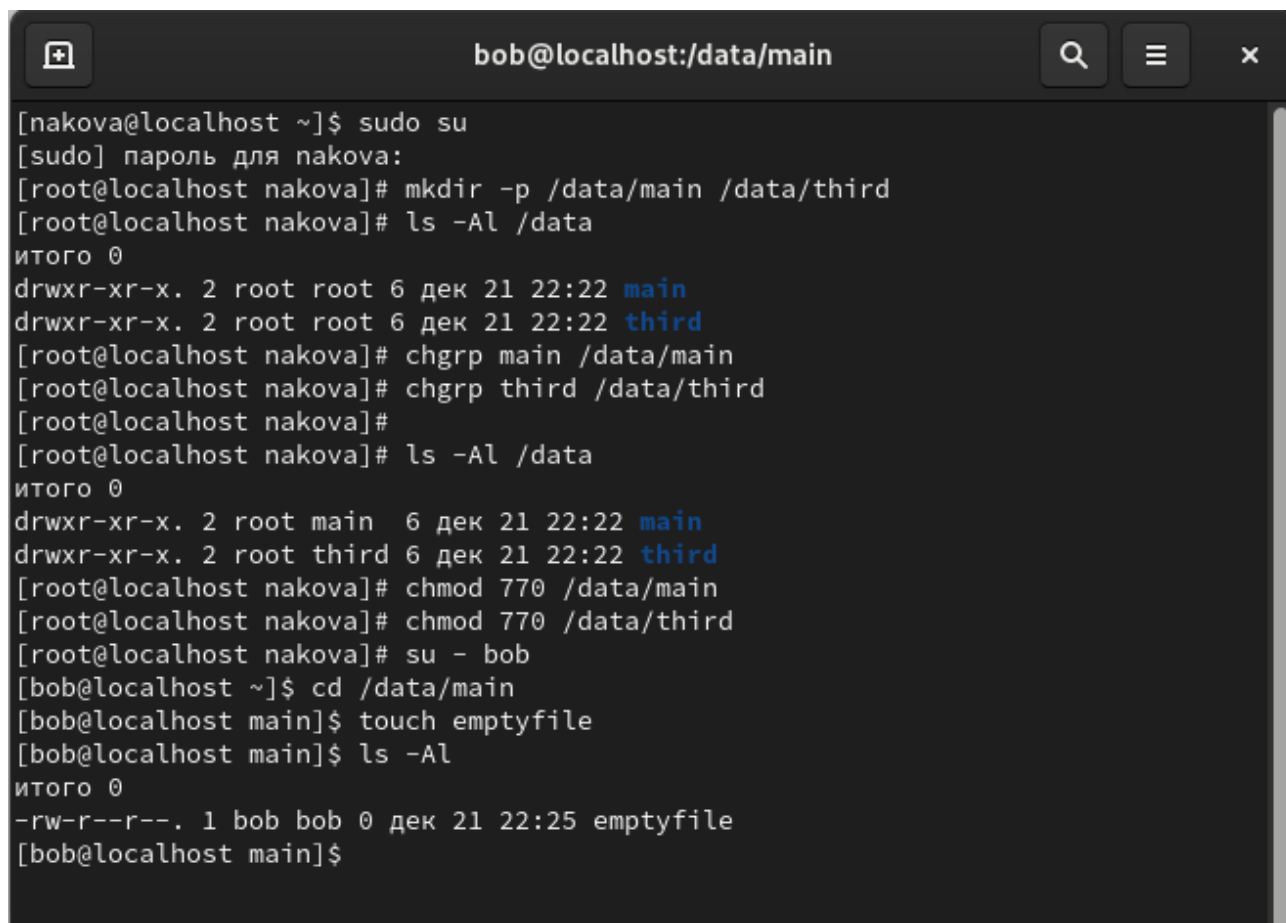
2022 г.

Цель работы:

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

Выполнение 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. 1):

A terminal window titled 'bob@localhost:/data/main'. The terminal shows a sequence of commands and their outputs. It starts with a user 'nakova' running 'sudo su' to become root. Then, 'mkdir -p /data/main /data/third' is run to create directories. 'ls -Al /data' shows the current state. Then 'chgrp main /data/main' and 'chgrp third /data/third' are run to change group ownership. Another 'ls -Al /data' shows the updated state. Then 'chmod 770 /data/main' and 'chmod 770 /data/third' are run to set permissions. 'su - bob' switches back to user 'bob'. Finally, 'cd /data/main' and 'touch emptyfile' are run, and a final 'ls -Al' shows the new file 'emptyfile' created in the 'main' directory.

```
[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. 1. Открытие учётной записи (su -), создание каталогов (mkdir -p), просмотр владельцев каталогов (ls -Al), изменение владельцев каталогов (chgrp), установка разрешений (chmod).

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):

```
rin ~]$ su - bob  
  
n]$ cd /data/main  
n]$ touch emptyfile  
n]$ ls  
  
n]$ cd /data/third  
ird: Permission denied  
n]$
```

Рис. 2. Открытие учётной записи пользователя bob, переход и проверка по каталогам main и third, создание новых файлов.

Откроем новый терминал под пользователем alice: **su - alice**. Перейдём в каталог /data/main: **cd /data/main**. В нём создадим два файла, владельцем которых является alice: **touch alice1** и **touch alice2**. Командой **ls** проверим корректность выполнения предыдущей команды (Рис. 3).

Рис. 3. Открытие учётной записи пользователя alice, переход в каталог main, создание двух файлов, проверка.

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. 4).

```
ls
ile
rm -f alice*
ls

touch bob1
touch bob2
```

Рис. 4. Проверка созданных файлов под пользователем bob, удаление файлов, создание двух новых файлов.

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. 5).

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

Рис. 5. Открытие терминала под пользователем root, установка бит идентификатора группы, а также sticky-бита для разделяемого (общего) каталога группы.

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. 6).

```

[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. 6. Открытие терминала под пользователем alice, создание в каталоге main двух новых файлов, проверка принадлежности файлов группе main и попытка удаление файлов пользователя bob.

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. 7.2).

```

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

```

Рис. 7.1. Открытие терминала с учётной записью root, установка прав на чтение и выполнение.

```

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. 7.2. Проверка правильности установки разрешений.

Далее создадим новый файл с именем `newfile1` в каталоге `/data/main`: **`touch /data/main/newfile1`**. Используем `getfacl /data/main/newfile1` для проверки текущих назначений полномочий. У пользователя только чтение и запись, у группы и других только чтение (Рис. 8).

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

Рис. 8. Создание нового файла и проверка текущих назначений полномочий.

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. 9.2).

```
# setfacl -m d:g:third:rwX /data/main
# setfacl -m d:g:main:rwX /data/third
# touch /data/main/newfile2
# getfacl /data/main/newfile2
```

Рис. 9.1. Установка ACL по умолчанию для двух каталогов, добавление нового файла в каталог `main` и проверка текущих назначений полномочий.

```
getfacl: Removing leading '/' from absolute path names
# file: data/third/newfile2
# owner: root
# group: root
user::rw-
group::rwx                               #effective:rw-
group:main:rwx                           #effective:rw-
mask::rw-
other::---
```

Рис. 9.2. Добавление нового файла в каталог third и проверка текущих назначений полномочий.

Для проверки полномочий группы third в каталоге /data/third войдём в другом терминале под учётной записью члена группы third: **su – carol** и проверим операции с файлами: **rm /data/main/newfile1** и **rm /data/main/newfile2**. Система не даёт удалить данные файлы. Теперь проверим, возможно ли осуществить запись в файл:

```
echo "Hello, world" >> /data/main/newfile1
```

```
echo "Hello, world" >> /data/main/newfile2
```

В файл newfile1 запись осуществить не получилось, а вот в newfile2 всё выполнилось (Рис. 10).

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

никаких прав для других? Приведите пример. **chmod 770 /data** (скриншот из лабораторной работы).

```
]# chmod 770 /data/main  
]# chmod 770 /data/third
```

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”** (скриншот из лабораторной работы).

```
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:----
```

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** (скриншот из лабораторной работы).

```
]# 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 <file/dir>** (скриншот из лабораторной работы).

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

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.**

Вывод:

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