Отчёт по лабораторной работе №3

Настройка прав доступа

Максат Хемраев

Содержание

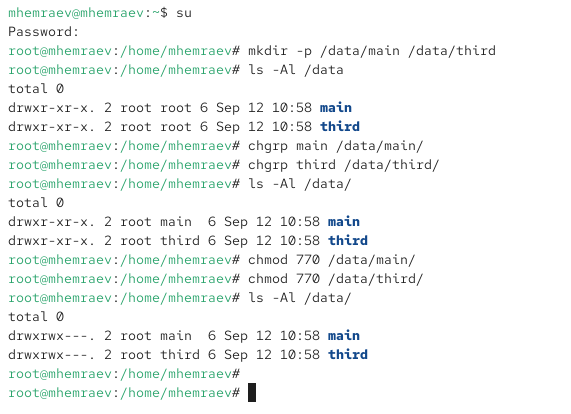
# 1 Цель работы

Получение навыков настройки базовых и специальных прав доступа для групп пользователей в операционной системе типа Linux.

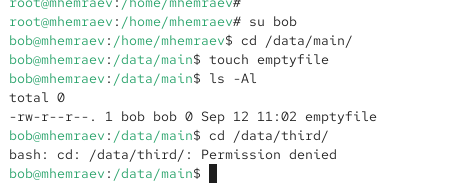
# 2 Отчёт по выполнению работы

## 2.1 Управление базовыми разрешениями

1. Вошёл в систему под пользователем **root**. Затем создал структуру каталогов */data/main* и */data/third*.  
   Проверка показала, что владельцем по умолчанию стал суперпользователь **root**, а владельцем группы — также **root**.

* 
* Рис. 1: Создание каталогов main и third

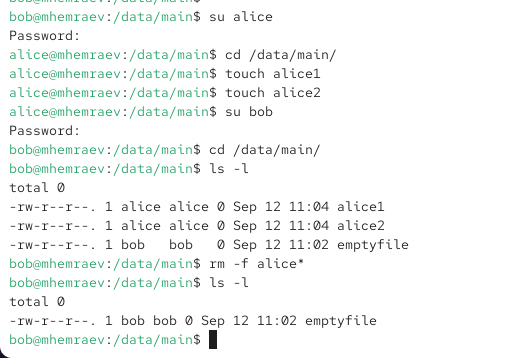
1. Изменил владельцев групп каталогов: для */data/main* установил группу **main**, а для */data/third* — группу **third**.  
   После проверки стало видно, что каталоги теперь принадлежат соответствующим группам.
2. Задал права доступа к каталогам так, чтобы только владельцы и их группы имели возможность чтения, записи и выполнения. Всем остальным пользователям доступ был запрещён. Для этого применил команды *chmod 770 /data/main* и *chmod 770 /data/third*.  
   Повторная проверка подтвердила корректное применение прав: теперь доступ имеют только владельцы и группы.
3. Вошёл в систему под пользователем **bob**. Перешёл в каталог */data/main* и создал там пустой файл **emptyfile**.  
   Проверка содержимого показала, что файл успешно создан, его владельцем стал пользователь **bob**, а права доступа ограничены для других.

* 
* Рис. 2: Создание файла emptyfile пользователем bob

1. После этого под пользователем **bob** была предпринята попытка перейти в каталог */data/third*. Доступ оказался запрещён, так как группа каталога установлена **third**, а пользователь **bob** в неё не входит. Соответственно, bob не имеет прав на выполнение операций в этом каталоге.

## 2.2 Управление специальными разрешениями

1. Вошёл в систему под пользователем **alice** и перешёл в каталог */data/main*.  
   Создал два файла — **alice1** и **alice2**, владельцем которых является alice.

* 
* Рис. 3: Создание файлов alice1 и alice2

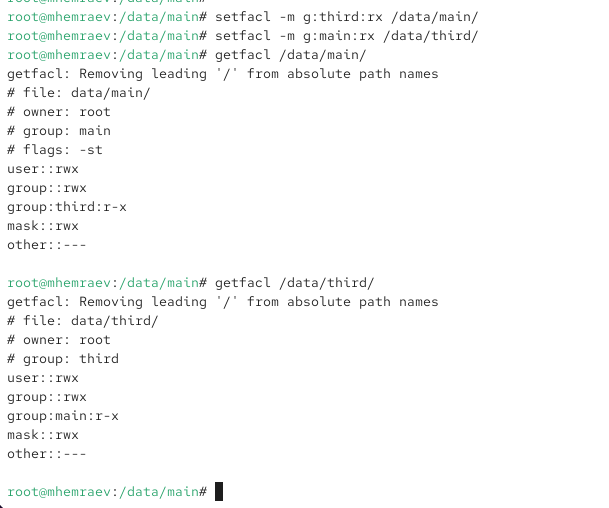
1. В другом терминале перешёл под учётную запись **bob**. В каталоге */data/main* выполнил просмотр содержимого и убедился в наличии файлов alice.  
   Затем удалил их, что оказалось возможным из-за отсутствия дополнительных ограничений.
2. Под пользователем **bob** создал в каталоге два собственных файла — **bob1** и **bob2**.
3. Далее под пользователем **root** для каталога */data/main* установил бит идентификатора группы и sticky-бит с помощью команды *chmod g+s,o+t /data/main*.  
   Это позволило назначить наследование группы **main** для всех создаваемых файлов, а также запретило удаление чужих файлов пользователями.
4. Вернувшись под пользователем **alice**, создал в каталоге ещё два файла — **alice3** и **alice4**. Проверка показала, что оба файла принадлежат группе **main**, которая является группой-владельцем каталога.

* 
* Рис. 4: Создание файлов alice3 и alice4 с установленными спецразрешениями

1. Под alice была предпринята попытка удалить файлы bob (**bob1** и **bob2**). Система отклонила запрос, выдав сообщение *Operation not permitted*. Sticky-бит сработал корректно, сохранив файлы bob от удаления другим пользователем.

## 2.3 Управление расширенными разрешениями с использованием списков ACL

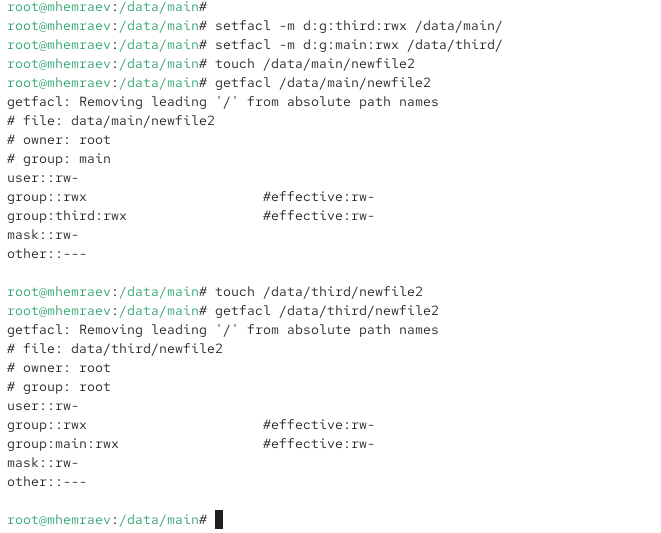
1. Вошёл в систему под пользователем **root**. Для каталога */data/main* установил права на чтение и выполнение для группы **third**, а для каталога */data/third* — права на чтение и выполнение для группы **main**.

* 
* Рис. 5: Установка ACL для групп main и third

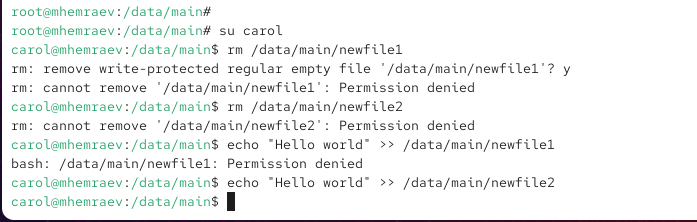
1. Проверил применённые разрешения с помощью команды *getfacl*. Каталоги отобразили новые записи для дополнительных групп.
2. Создал файл **newfile1** в каталоге */data/main*. Проверка с помощью *getfacl* показала, что права доступа у файла остались стандартными: владелец имеет доступ к чтению и записи, группа получила только права на чтение. Это объясняется тем, что новые ACL были назначены только на каталог, а не как наследуемые по умолчанию для новых файлов.  
   Аналогичные действия выполнил в каталоге */data/third* и получил тот же результат.

* 
* Рис. 6: Создание и проверка файла newfile1

1. Установил ACL по умолчанию: для группы **third** в каталоге */data/main* — *rwx*, а для группы **main** в каталоге */data/third* — *rwx*.
2. Создал новые файлы **newfile2** в каталогах */data/main* и */data/third*. Теперь при проверке с помощью *getfacl* было видно, что права по умолчанию успешно применились: группы **third** и **main** получили доступ к чтению и записи.

* 
* Рис. 7: Создание и проверка файла newfile2

1. Для проверки полномочий вошёл в систему под пользователем **carol**, который входит в группу **third**.  
   Попытка удалить файлы **newfile1** и **newfile2** завершилась ошибкой *Permission denied*, что подтвердило ограничение на удаление.  
   Также при попытке записи данных в файл **newfile1** система отказала, так как у него не были заданы наследуемые ACL. Однако в файл **newfile2** запись оказалась возможна, так как он был создан уже после установки ACL по умолчанию.

* 
* Рис. 8: Проверка работы ACL под пользователем carol

# 3 Контрольные вопросы

### 3.0.1 1. Как следует использовать команду chown, чтобы установить владельца группы для файла? Приведите пример.

* Команда chown позволяет изменить владельца и/или группу файла.
* Синтаксис: chown :<имя\_группы> <файл> — изменяет только группу.
* Пример:

chown :main /data/file1

### 3.0.2 2. С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример.

* Для поиска используется команда find.
* Пример:

find /home -user alice

### 3.0.3 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.

* Используется команда chmod.
* Пример:

chmod -R 770 /data

### 3.0.4 4. Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?

* Используется команда chmod +x.
* Пример:

chmod +x script.sh

### 3.0.5 5. Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример.

* Для этого применяется **SGID-бит**.
* Пример:

chmod g+s /data/main

### 3.0.6 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.

* Для этого используется **sticky-бит**.
* Пример:

chmod +t /data/main

### 3.0.7 7. Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге?

* Используется команда setfacl.
* Пример:

setfacl -m g:main:r ./\*

### 3.0.8 8. Что нужно сделать для гарантии того, что члены группы получат 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.

* Нужно применить ACL с опцией наследования -d.
* Пример:

setfacl -R -m g:main:r /data  
setfacl -R -d -m g:main:r /data

### 3.0.9 9. Какое значение umask нужно установить, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы? Приведите пример.

* Нужно установить umask 007.
* Пример:

umask 007

### 3.0.10 10. Какая команда гарантирует, что никто не сможет удалить файл myfile случайно?

* Нужно убрать права на запись и установить атрибут защиты от удаления.
* Пример:

chmod a-w myfile  
chattr +i myfile

# 4 Заключение

В ходе работы были изучены и применены базовые, специальные и расширенные разрешения в Linux. Создавались каталоги, назначались права доступа, использовались sticky-бит, SGID и ACL, что позволило реализовать разграничение доступа и обеспечить безопасное совместное использование ресурсов пользователями разных групп.