Отчёт по лабораторной работе №3

Настройка прав доступа

Эзиз Хатамов

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

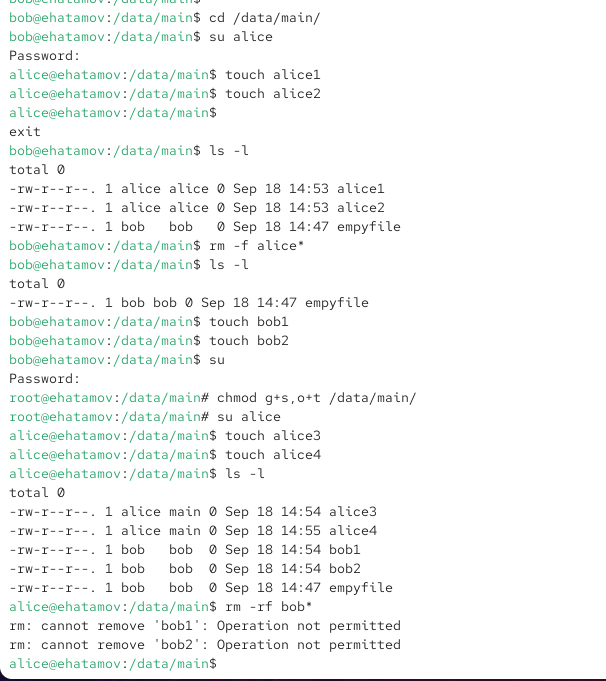
## 2.1 Управление базовыми разрешениями

1. В корневом каталоге были созданы каталоги **/data/main** и **/data/third**.  
   Первоначально владельцем обоих каталогов являлся пользователь **root**.
2. Владельцы каталогов были изменены на соответствующие группы:
   * каталог **/data/main** — группа **main**,
   * каталог **/data/third** — группа **third**.  
     Проверка командой ls -Al /data подтвердила корректность изменений.
3. Для каталогов были установлены права доступа 770, что позволяет владельцу и членам группы читать, записывать и выполнять действия с содержимым, при этом все остальные пользователи лишены доступа.
4. Под пользователем **bob** была предпринята попытка войти в каталог **/data/main** и создать файл *emptyfile*.  
   Операция прошла успешно, так как пользователь **bob** является участником группы **main** и имеет права на запись в данный каталог.
5. Затем пользователь **bob** попробовал перейти в каталог **/data/third**.  
   В доступе было отказано, так как он не входит в группу **third** и не имеет прав на чтение или выполнение в этом каталоге.

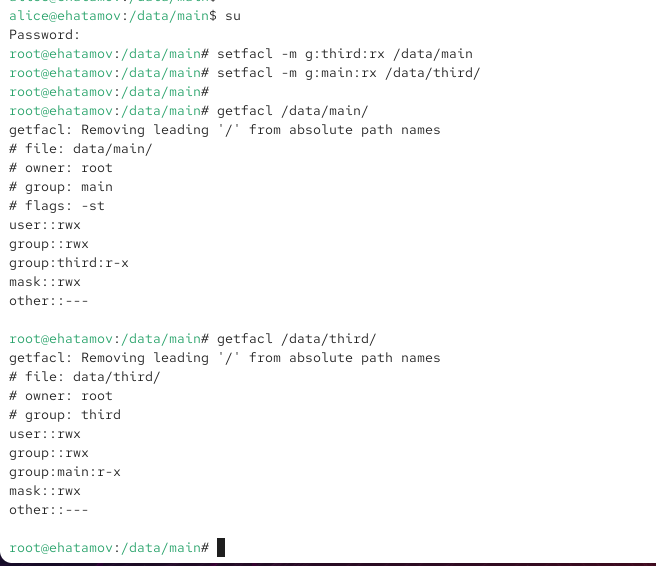
* 
* Рис. 1: Управление доступом к каталогам /data/main и /data/third

## 2.2 Управление специальными разрешениями

1. В каталоге **/data/main** под пользователем **alice** были созданы файлы *alice1* и *alice2*.
2. Под пользователем **bob** выполнен переход в каталог **/data/main**.  
   Пользователь смог просмотреть содержимое каталога и удалить файлы, принадлежащие **alice**, так как sticky-бит ещё не был установлен.
3. Под пользователем **bob** были созданы файлы *bob1* и *bob2*.
4. Под пользователем **root** для каталога **/data/main** были установлены:
   * бит идентификатора группы (**setgid**),
   * sticky-бит (**+t**).  
     Это обеспечило автоматическое наследование групповой принадлежности новых файлов и запрет на удаление файлов другими пользователями.
5. После этого под пользователем **alice** в каталоге **/data/main** были созданы файлы *alice3* и *alice4*.  
   Их групповым владельцем стала группа **main**, что подтверждает корректное действие **setgid**.
6. Попытка удалить файлы *bob1* и *bob2*, принадлежащие пользователю **bob**, завершилась неудачно.  
   Это произошло благодаря установленному **sticky-биту**, который запрещает удаление файлов, владельцем которых является другой пользователь.

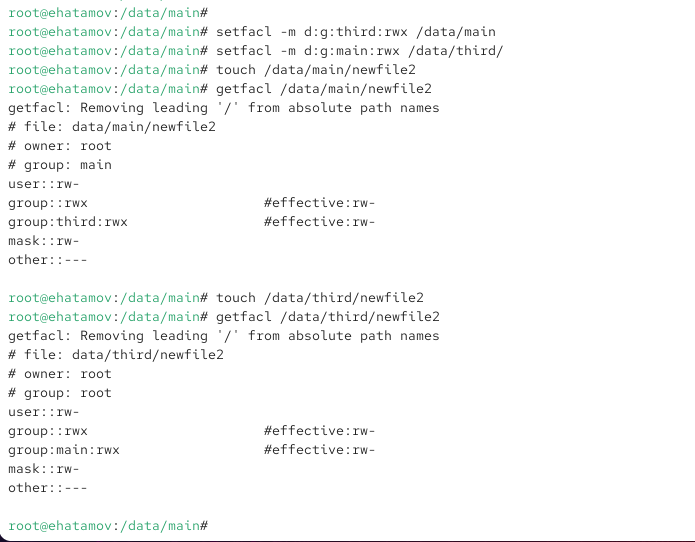
* 
* Рис. 2: Управление специальными разрешениями в каталоге /data/main

## 2.3 Управление расширенными разрешениями с использованием списков ACL

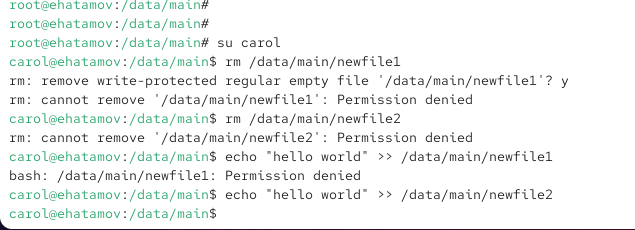
1. В каталоге **/data/main** для группы **third** были назначены права на чтение и выполнение, а в каталоге **/data/third** для группы **main** — аналогичные права.  
   Проверка с помощью getfacl подтвердила корректность изменений:
   * у каталога **/data/main**: группа *third* имеет r-x,
   * у каталога **/data/third**: группа *main* имеет r-x.
   * 
   * Рис. 3: Проверка управления расширенными разрешениями ACL
2. В каталоге **/data/main** был создан файл *newfile1*.  
   Проверка через getfacl показала, что у файла права доступа только для владельца (**root**), а группа и другие пользователи имеют доступ на чтение.  
   Это связано с тем, что права ACL на сам каталог не наследуются автоматически на новые файлы без установки **ACL по умолчанию**.

* Аналогичная ситуация наблюдается в каталоге **/data/third**: созданный там файл *newfile1* также имеет стандартные права без учёта ACL каталога.
* 
* Рис. 4: Создание файлов и проверка ACL

1. Для каталогов были назначены **ACL по умолчанию**:
   * для каталога **/data/main** группе **third** назначен доступ rwx,
   * для каталога **/data/third** группе **main** назначен доступ rwx.
2. После этого при создании новых файлов (*newfile2*) в каталогах /data/main и /data/third, права доступа унаследовали настройки по умолчанию:
   * файлы в /data/main получили права rw- для группы **third**,
   * файлы в /data/third получили права rw- для группы **main**.

* 
* Рис. 5: Установка ACL по умолчанию и проверка наследования

1. Проверка доступа под пользователем **carol** (член группы **third**) показала:
   * удалить файлы *newfile1* и *newfile2* в каталоге **/data/main** не удалось — права ACL не предоставляют возможность удаления чужих файлов,
   * выполнить запись в эти файлы также не удалось.  
     Таким образом, даже при наличии ACL rw- для группы, операция блокируется, если права владельца и маска ACL запрещают действие.

* 
* Рис. 6: Проверка доступа пользователем carol

# 3 Контрольные вопросы

## 3.1 1. Укажите команды терминала и приведите примеры

1. **Как следует использовать команду chown, чтобы установить владельца группы для файла? Приведите пример.**
   * chown :<группа> <файл> — установить владельца группы.
   * Пример: chown :developers script.sh — назначает файл *script.sh* группе **developers**.
2. **С помощью какой команды можно найти все файлы, принадлежащие конкретному пользователю? Приведите пример.**
   * find / -user <имя\_пользователя>
   * Пример: find /home -user alice — ищет все файлы в каталоге */home*, принадлежащие пользователю **alice**.
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 -R 770 /data
   * Пример: chmod -R 770 /data — пользователи и группы получают полный доступ, остальные не имеют прав.
4. **Какая команда позволяет добавить разрешение на выполнение для файла, который необходимо сделать исполняемым?**
   * chmod +x <файл>
   * Пример: chmod +x script.sh — делает *script.sh* исполняемым.
5. **Какая команда позволяет убедиться, что групповые разрешения для всех новых файлов, создаваемых в каталоге, будут присвоены владельцу группы этого каталога? Приведите пример.**
   * chmod g+s <каталог> — установка **setgid** на каталог.
   * Пример: chmod g+s /data/main — все новые файлы и подкаталоги в */data/main* будут наследовать групповую принадлежность.
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.**
   * Установить **sticky-бит** на каталог:
   * chmod +t <каталог>
   * Пример: chmod +t /tmp — в этом каталоге пользователь может удалять только свои файлы.
7. **Какая команда добавляет ACL, который предоставляет членам группы права доступа на чтение для всех существующих файлов в текущем каталоге?**
   * setfacl -m g:<группа>:r <файл>
   * Пример: setfacl -m g:developers:r \* — даёт группе **developers** права чтения на все файлы в текущем каталоге.
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.**
   * Использовать **ACL по умолчанию**:
   * setfacl -R -m g:<группа>:r <каталог>
   * setfacl -d -m g:<группа>:r <каталог>
   * Пример:
   * setfacl -R -m g:developers:r /data/projects   
     setfacl -d -m g:developers:r /data/projects
9. **Какое значение umask нужно установить, чтобы «другие» пользователи не получали какие-либо разрешения на новые файлы? Приведите пример.**
   * umask 007 — запрещает права для категории “others”.
   * Пример: umask 007 → новые файлы будут создаваться с правами rw-rw----.
10. **Какая команда гарантирует, что никто не сможет удалить файл myfile случайно?**

* chattr +i myfile — делает файл неизменяемым.
* Для проверки: lsattr myfile.
* Чтобы снять атрибут: chattr -i myfile.

# 4 Заключение

В ходе работы были изучены основы управления пользователями и группами в Linux, а также методы настройки прав доступа и параметров паролей.