# Лабораторная работа №5

## Работа с пользователями. Управление правами доступа. Управление файлами и каталогами. Ссылки.

Цель работы: Изучить возможности Linux при работе с пользователями и управлении правами доступа.

Задачи работы:

* Рассмотреть концепцию Linux при работе с пользователями;
* Изучить управление базами данных пользователей;
* Рассмотреть возможности манипулирования доступом к данным.

**Порядок выполнения работы**

* **Подготовьте свою виртуальную машину, на которой будете выполнять лабораторную работу**
* **Запустите Oracle VM VirtualBox, импортируйте машину Ubuntu Server01**
* **Пока идёт импорт (около 15 минут) ознакомьтесь с теоретической частью**
* **Запустите Ubuntu Server01, войдите в него с логином и паролем администратора, создайте пользователя с именем – вашей фамилией, включите его в группу sudo. Перезагрузите сервер и войдите под вашим логином.**
* **Выполните все действия, описанные в теоретической части (для тренировки).**
* **Выполните задания, изложенные в конце данных методических указаний, сохраните их при помощи скриншотов и включите в отчёт по лабораторной работе.**
* **Ответьте письменно на два любых контрольных вопроса, ответы также включите в отчёт.**
* **Сохраните отчёт в общей папке на сервере, для проверки, и на своём носителе.**
* **Экспортируйте образ сервера на свой носитель или в своё** **облако для последующей работы.**

Теоретическая часть

Работа с пользователями

Linux - это многопользовательская операционная система. Каждый пользователь в Linuxпринадлежит одной основной группе и одной или нескольким дополнительнымгруппам. В Linux, как и в большинстве других операционных системах, работа с пользователями заключается в наборе следующих манипуляций: добавление пользователя/группы, удаление пользователя/группы, модификация настроек пользователя/группы. Данные манипуляции производятся с помощью команд: useradd, groupadd, userdel, groupdel, usermod, groupmod, а так же passwd, gpasswd, id. Существуют так же и графические средства администрирования пользователями, обычно они расположены в оболочке X в разделеАдминистрирование - Пользователи и группы. Однако, при администрировании Linux использование графических оболочек не приветствуется.

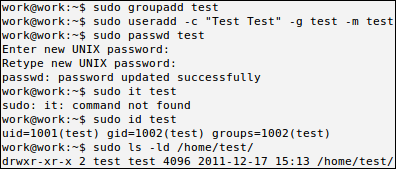
UID, GID

Каждый пользователь в системе имеет свой уникальный идентификационный номер (user-ID, или UID). Также пользователи могут объединяться в группы, которые в свою очередь имеют group-ID, или GID. Чтобы узнать свой UID и GID, т.е. уникальный номер пользователя и номер группы, к которой вы принадлежите, необходимо ввести команду id (рис. 5.1).



*Рис. 5.1. Отобразить UID и GID*

Пример добавления пользователя (рис. 5.1.):



*Рис. 5.2. Добавление нового пользователя*

В примере мы добавляем группу для нового пользователя (groupadd), далее создаем нового пользователя с полным именем Test Test, имеющего основную группу test и логин test, далее задаем пароль для пользователя test (passwd test) и проверяем параметры созданного пользователя (id и созданный каталог пользователя /home/test/). На рис. 5.2 видно, что UID и GID - более 1000. Данная особенность является признаком обычного пользователя. Значения ниже (меньше) 1000 (а в некоторых версиях - меньше 500) указывают на то, что пользователь является системным пользователем.

В соответствии с соглашением, системные пользователи обычно имеют id меньше, чем 100, а пользователь rootимеет id, равный 0. Автоматическая нумерация обычных пользователей начинается со значения UID\_MIN, установленного в файле /etc/login.defs. Это значение обычно установлено в 500 или 1000.

Помимо учетных записей обычных пользователей и учетной записи пользователя root, в системе бывает несколькоучетных записей специального назначения для демонов, таких как FTP, SSH, mail, news и т.д. Такие учетные записи часто управляют файлами, но к ним невозможно получить доступ путем обычной регистрации в системе. Поэтому обычно они имеютlogin shell, определенный как /sbin/nologin или/bin/false, чтобы попытки зарегистрироваться в системе терпели неудачу.

Управление базами данных пользователей и групп в Linux

Основные файлы, содержащие информацию о пользователях и группах, - это четыре файла в каталоге */etc.*

1. /etc/passwd - файл паролей, содержащий основную информацию о пользователях;
2. /etc/shadow - файл теневых шифрованных паролей, содержащий зашифрованные пароли;
3. /etc/group - файл групп, содержащий основную информацию о группах и принадлежащих этим группам пользователях;
4. /etc/gshadow - файл теневых групп, содержащий шифрованные пароли групп.

Данные файлы редактировать обычным текстовым редактором крайне не рекомендуется. Они, обновляются при выполнении вышеуказанных команд, при этом при изменении - блокируются и синхронизируются.

Если все же есть острая необходимость в редактировании указанный файлов, то при помощи команды vipw можно безопасно редактировать файл/etc/passwd, а при помощи команды vigr безопасно редактировать файл /etc/group. Эти команды заблокируют необходимые файлы на то время, пока при помощи редактора vi будут производиться изменения. Если вы вносите изменения в файл/etc/passwd, команда vipwподскажет, что необходимо проверить, не нужно ли обновить и файл /etc/shadow. Подобным образом, если вы обновляете файл /etc/group при помощи команды vigr, вы получите подсказку, что необходимо обновить и файл /etc/gshadow. Если необходимо удалить администраторов группы, необходимо использовать команду vigr, поскольку команда gpasswdпозволяет только добавлять администраторов.

В современных системах, файлы passwd и group не хранят пароли в открытом виде. Это сделано из соображений безопасности. Сами файлы passwd и group должны быть доступными для чтения для всех, а зашифрованные пароли - недоступными для чтения для всех. Поэтому зашифрованные пароли хранятся в теневых файлах, и эти файлы доступны для чтения только пользователю root. Необходимый доступ для изменения аутентификационных данных обеспечивается при помощи suid-программы, которая имеет полномочия пользователя root, но может быть запущена любым пользователем.

Права доступа

Операционная система Linux - это многопользовательская система, которая дает огромные возможности манипулирования доступом к данным для каждого пользователя отдельно. Это позволяет гибко регулировать отношения между пользователями, объединяя их в группы, что позволит защитить данные одного пользователя от нежелательного вмешательства других.

Бессмысленно считать, что файловая система это не самая важная часть операционной системы, поскольку все данные пользователей хранятся именно в файлах.

В UNIX-подобных системах файлы также обеспечивают доступ к периферийным устройствам, дисковым накопителям, принтерам и т.п.

Права доступа к файлам

В свою очередь файлы имеют двух владельцев: пользователя (user owner) и группу пользователей (group owner). Для каждого файла есть индивидуальные права доступа, которые разбиты на три группы:

* Доступ для пользователя-владельца файла (owner).
* Доступ для группы-владельца файла (group).
* Доступ для остальных пользователей (others).

Для каждой категории устанавливаются три вида доступа: (x) - право на запуск файла, (r) - право на чтение файла, (w) - право на изменение (редактирование) файла.

Для того, чтобы увидеть права доступа к файлам необходимо ввести команду ls с ключом –l (рис. 5.3).



*Рис. 5.3. Просмотр прав доступа к файлу*

Для данного примера мы видим, что владелец имеет права на чтение, запись (первые две буквы rw), группа пользователей может лишь читать этот файл (следующая r--), остальные пользователи могут также только читать данный файл (r--).

Изменение прав доступа

Права пользователя могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда:

chmod[u|g|o|a] [+|-|=] [r|w|x] name1 [name2 ...]

В качестве аргументов команда принимает указание классов доступа («u» - владелец-пользователь, «g» - владелец-группа, «o» - остальные пользователи, «a» - все вышеперечисленные группы вместе), права доступа («r» - чтение, «w» - запись, «x» - выполнение) и операцию, которую необходимо произвести («+» - добавить, «-» -убрать, «=» - присвоить).

Таким образом, чтобы разрешить выполнение файла ip, который находиться в директории /home/work/Загрузки всем пользователем необходимо выполнить команду (рис. 5.4):



*Рис. 5.4. Команда, выдающая права на исполнение файла*

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить (рис.5.5):



*Рис. 5.5. Команда, повзоляющая оставить права записи только для владельца файла*

Рассмотрим еще несколько примеров:

* chmod go=w ip - установить право на запись для всех пользователей кроме владельца;
* chmod a+x ip - предоставить право на запись для всех пользователей;
* chmod g+x-w ip - добавить для группы право на выполнения файла, но снять право на запись.

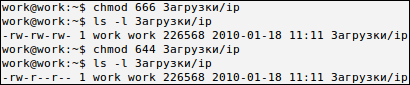
Права доступа можно представить в виде битовой строки, в которой каждые 3 бита определяют права доступа для соответствующей категории пользователей, как представлено в таблице 5.1:

Таблица 5.1

Представление прав доступа в виде битовой строки

|  |  |  |
| --- | --- | --- |
| rwx | rwx | rwx |
| 421 | 421 | 421 |
| user | group | others |
| владелец | группа | остальные |

Таким образом, для команды chmod 666 ip имеем (рис. 5.6):



*Рис. 5.6. Пример испоьлзования команды chmod*

Команда:

chmod 644 имя\_файла

устанавливает «обычные» права доступа, т.е. владелец может читать и записывать в файл, а все остальные пользователи - только читать.

Особенности прав доступа для каталогов

Права доступа для каталогов не столь очевидны. Это в первую очередь связано с тем, что система трактует операции чтения и записи для каталогов отлично от остальных файлов. Право чтения каталога позволяет Вам получить имена (и только имена) файлов, находящихся в данном каталоге. Чтобы получить дополнительную информацию о файлах каталога (например, подробный листинг команды ls -l), системы придется «заглянуть» в метаданные файлов, что требует права на выполнения для каталога. Право на выполнение также потребуется для каталога, в который Вы захотите перейти (т.е. сделать его текущим) с помощью команды cd.

T-бит, SUID и SGID

Помимо стандартных «rwx» значений существуют еще и буквы «s» и «t». В действительности, битовая маска прав доступа к файлам содержит 4 группы по 3 бита в каждой. Таким образом, команда chmod 755 это всего лишь краткая запись полной формы команды: chmod 0755.

T-бит обычно используется с каталогами. Обычно, когда t-бит для каталога не установлен, файл в данном каталоге может удалить любой пользователь, имеющий доступ на запись к данному файлу. Устанавливая t-бит на каталог мы меняем это правило таким образом, что удалить файл из каталога может только владелец этого каталога или файла.

Установить t-бит можно при помощи команд:

chmod a+tw имя\_файла

chmod 1777 имя\_файла

Атрибуты SUID и SGID позволяют изменить права пользователя при запуске на выполнения файла, имеющего эти атрибуты.

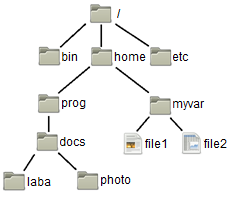
Запускаемая программа получает права доступа к системным ресурсам на основе прав доступа пользователя, запустившего программу. Установка же флагов SUID и SGID изменяет это правило таким образом, что назначает права доступа к системным ресурсам исходя из прав доступа владельца файла. Т.е. запущенный исполняемый файл, которым владеет суперпользователь, получает права доступа к системным ресурсам на уровне суперпользователя (фактически неограниченные). При этом установка SUID приведет к наследованию прав владельца-пользователя файла, а установка SGID -владельца-группы.

Пользоваться такими мощными атрибутами как SUID и SGID нужно с крайней осторожностью, особенно подвергать пристальному вниманию программы и скрипты, владельцем которых является root (суперпользователь), т.к. это потенциальная угроза безопасности системы.

Управление файлами

В ОС Linux следует различать физическую файловую систему, которая отвечает за управление дисковым пространством и размещение файлов в физических адресах диска и логическую файловую систему, которая обеспечивает логическую структуру хранения файлов - пространство имен файлов. ОС Unix и Linux могут работать с различными физическими файловыми системами (Ext2, ext3, ufs), логическое же представление файловой системы в Unix/Linux структурировано. Все файлы в логической файловой системе располагаются в виде дерева, промежуточные вершины которого соответствуют каталогам, и листья - файлам и пустым каталогам. Реально на каждом логическом диске (разделе физического дискового пакета) располагается отдельная иерархия каталогов и файлов. Для получения общего дерева в динамике используется «монтирование» отдельных иерархий к фиксированной корневой файловой системе в качестве ветвей общего дерева. Самым верхом иерархии является корень, который имеет предопределенное имя «/» (слэш). Этот же символ используется как разделитель имен в пути. Далее в корне находятся папки с определенными для каждого дистрибутива именами (etc, home, bin, mnt, proc и т.д.).

Полное имя файла, например, /bin/sh означает, что в корневом каталоге должно содержаться имя каталога bin, а в каталоге bin должно содержаться имя файла sh. Коротким или относительным именем файла называется имя, задающее путь к файлу от текущего рабочего каталога. В каждом каталоге содержатся два специальных имени, имя «.» - ссылка на текущий каталог, и имя «...» - ссылка «родительский» каталог данного текущего каталога, т.е. каталог, непосредственно предшествующий данному в иерархии каталогов. Так, например, для структуры, показанной на рис. 5.7 доступ к файлу file2 из текущего каталога (laba) возможен по полному имени: /home/myvar/file2 или по относительному имени: .../.../.../myvar/file5.



*Рис. 5.7. Пример дерева каталогов*

Типы файлов

ОС LINUX поддерживают несколько типов файлов:

* Обычные файлы (или регулярные) - представляют собой последовательность байтов. Это текстовые, исполняемые файлы и т.д. Данный тип файла отображается командой ls -l в виде «-» (черточки).
* Каталоги - представляют собой особый вид файлов, которые хранятся во внешней памяти подобно обычным файлам, но их структура поддерживается самой файловой системой. Данный тип файла отображается командой ls -l в виде символа «d».
* Специальные файлы устройств, бывают блочные и символьные. Данный тип файла отображается командой ls -l в виде символа «b» или «с» соответственно. Специальные файлы не хранят данные. Они обеспечивают механизм отображения физических внешних устройств в имена файлов файловой системы. Каждому устройству, поддерживаемому системой, соответствует, по меньшей мере, один специальный файл. При выполнении чтения или записи по отношению к специальному файлу, производится прямой вызов соответствующего драйвера устройства. При этом имена специальных файлов можно использовать практически всюду, где можно использовать имена обычных файлов.
* Ссылка (link). Данный тип файла отображается командой ls -l в виде символа «l». Файловая система UNIX/LINUX обеспечивает возможность связывания одного и того же файла с разными именами.
* Именованный программный канал (pipe) - одно из средств межпроцессных взаимодействий (IPC) в ОС UNIX/LINUX. Данный тип файла отображается командой ls -l в виде символа «p». Именованному программному каналу обязательно соответствует элемент некоторого каталога.
* Сокет (socket)- предоставляют весьма мощный и гибкий IPC. Данный тип файла отображается командой ls -l в виде символа «s». Они могут использоваться для организации взаимодействия программ на одном компьютере, по локальной сети или через Internet, что позволяет создавать распределённые приложения различной сложности. Кроме того, с их помощью можно организовать взаимодействие с программами, работающими под управлением других операционных систем.

Ссылки

Существуют жесткие и мягкие ссылки.

Жесткая ссылка является просто еще одним именем для исходного файла и не является типом файла. Она прописывается в индексном дескрипторе исходного файла (в структуре, хранящей метаданные файла). После создания жесткой ссылки невозможно различить, где исходное имя файла, а где ссылка. Если вы удаляете один из этих файлов (точнее одно из этих имен), то файл еще сохраняется на диске (пока у него есть хоть одно имя - жесткая ссылка). Очень трудно различить первоначальное имя файла и позже созданные жесткие ссылки на него. Поэтому жесткие ссылки применяются там, где отслеживать различия и не требуется. Одно из применений жестких ссылок состоит в том, чтобы предотвратить возможность случайного удаления файла. Особенностью жестких ссылок является то, что они прямо указывают на номер индексного дескриптора, а, следовательно, такие имена могут указывать только на файлы внутри той же самой файловой системы (т. е., на том же самом носителе, на котором находится каталог, содержащий это имя).

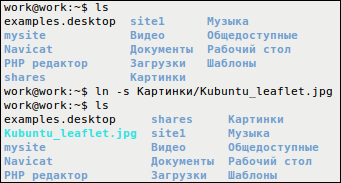
Мягкие (символические) ссылки тоже могут рассматриваться как дополнительные имена файлов, но в то же время они представляются отдельными файлами - файлами типа мягких ссылок и являются самостоятельным типом файла. Однако блоки данных файла в системе представляются в одном экземпляре, у файла-ссылки адреса блоков данных те же, что и у исходного файла. В отличие от жестких ссылок мягкие ссылки могут указывать на файлы, расположенные в другой файловой системе, например, на монтируемом носителе, или даже на другом компьютере. Если исходный файл удален, мягкая ссылка не удаляется, но становится бесполезной. Используйте мягкие ссылки в тех случаях, когда хотите избежать путаницы, связанной с применением жестких ссылок.

Создание любой ссылки внешне подобно копированию файла, но фактически как исходное имя файла, так и ссылка указывают на один и тот же реальный файл на диске. Поэтому, например, если вы внесли изменения в файл, обратившись к нему под одним именем, вы обнаружите эти изменения и тогда, когда обратитесь к файлу по имени-ссылке.

Для создания ссылки, используется команда ln (рис. 5.8):

ln [-f] файл1 [файл2 ...] целевой\_файл

Команда ln делает целевой\_файл ссылкой на файл1. Файл1 не должен совпадать с целевым\_файлом. Если целевой\_файл является каталогом, то в нем создаются ссылки на файл1, файл2,... с теми же именами. Только в этом случае можно указывать несколько исходных файлов. Если целевой\_файл существует и не является каталогом, его старое содержимое теряется. Аргументы: -f - удаление существующего целевого файла; -s – создание мягкой ссылки (по умолчанию создается жесткая ссылка).



*Рис. 5.8. Пример создания ссылки*

Файловый менеджер

Управление файлами также можно выполнять с помощью Midnight Commander (mc) - один из файловых менеджеров с текстовым интерфейсом типа Norton Commander для UNIX-подобных операционных систем. Запуск mc из консоли выполняется с помощью команды mc.

Установить файловый менеджер mc так, как показано на рис. 5.9 , делать этого не надо, он уже установлен в виртуальной машине Ubuntu Server01.



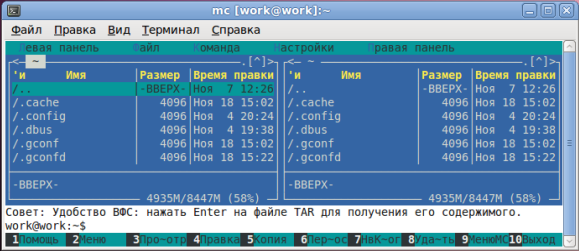
*Рис. 5.9. Установка mc*

Достоинство mc том, что есть встроенные средства редактирования и просмотра текстовых файлов (какими являются конфигурационные файлы). При работе с mc необходимы права [суперпользователя](http://inflin.narod.ru/srvubn/srv02.htm). Общая последовательность действий по редактированию конфигурационного файла:

* запустить программу (ввод команды mc);
* используя клавиши управления курсором и клавишу «Enter», добраться до нужного файла и выбрать его;
* нажатием клавиши «F4» открыть файл для редактирования;
* внести необходимые изменения;
* сохранить их (клавиша «F2»);
* выйти из режима редактирования (клавиша «F10»).

Просмотр файла выполняется аналогично, только клавишей «F3».

Выйти из mc можно тоже клавишей «F10». Общий вид mc приведен на рис. 5.10.



*Рис. 5.10. Midnight Commander*

**Задания к лабораторной работе:**

1. Выведите на экран UID и GID своего пользователя;
2. Создайте группу пользователей с именем usersGroup;
3. Создайте пользователя myUser в группе usersGroup;
4. Выведите на экран UID и GID пользователя myUser;
5. В своем пользователе work создайте текстовый файл, и ограничьте к нему доступ на чтение для всех других групп пользователей. Затем, зайдите в систему от имени пользователя myUser и проверьте возможность открыть этот текстовый файл;
6. От имени пользователя work изменить права доступа к данному файлу, и проверьте изменения;
7. Для одного и тоже файла создайте мягкую и жесткую ссылку в домашнем каталоге. Попробуйте создать ссылки одновременно для нескольких файлов;
8. Проверьте работу файловог менеджера mc.

Контрольные вопросы

1. Расскажите про идентификационные номера пользователей и групп в Linux.
2. Расскажите о файлах Linux, содержащих информацию о пользователях и группах системы.
3. Как система Linux хранит пароли пользователей и групп?
4. Как организуется разграничение доступа к файлам в Linux?
5. Как изменить права доступа к файлу? Как сделать это с помощью битовой строки?
6. Расскажите о Т-бит, SUID и SGID.
7. Расскажите про файловые системы Linux.
8. Какие типы файлов существуют в Linux?
9. Расскажите о мягких и жестких ссылках.