**Лабораторная работа №4**

**Основы администрирования на Linux**

К системному администрированию относится все то, что позволяет поддерживать систему в работоспособном и безопасном состоянии.

**Типы файлов**

**В Linux тип файла не опре­деляется по его расширению как в Windows.** Хотя маркировка файлов с по­мощью точки и трех букв допускается и используется в **Linux**, особенно для стандартных файлов, создаваемых различными приложениями, но это толь­ко для удобства пользователей. Например, для файлов рисунков и в **Linux** используют маркировку типа: *jpg*, *gif*, *tiff*, *wmf*, а для текстовых файлов ши­роко используется метка *txt*. Приведем классификацию файлов в **Linux**:

* ***обычные файлы или рядовые файлы***(*regular*). Такие файлы могут содержать различные данные, аналогично файлам в **MS-DOS**, в том числе и двоичный код программ. В программе **Midnight Commander** они отображаются белым и серым цветом. Если файл отно­сится к исполняемому типу (программе), то он выделяется зеленым цве­том, а перед именем ставится символ «**\***».
* ***каталоги***(*directory*). Эти файлы предназначены для хранения списков файлов и подкаталогов. Имена каталогов при отображении в **Midnight Commander** предваряются символом "**/**".
* ***файлы устройств***(*devices*). Имена таких файлов можно найти в каталоге **/dev**, они составляются, например, из двух номеров: номера класса уст­ройства и порядковый номер в данном классе. Напрямую с этими файлами пользователи дела не имеют, разве что когда приходит­ся в настройках выбирать номер СОМ-порта.
* *специальные файлы*: ***сокеты***(*sockets*) и ***именованные каналы***(*named* *pipes*). Такие файлы предназначены для обмена данными между процес­сами и интересны только для разработчиков. В программе **Midnight Commander** файлы устройств выделяются фиолетовым цветом, а перед именем добавляется символ «**-**».
* ***символические ссылки***(*symlinks*). Эти файлы наиболее близки к понятию ярлыков в **Windows**. To есть основное назначение символических ссылок в том, чтобы не копировать каждый раз файл в тот или иной каталог, а указывать его «адрес» в файле символической ссылкой. При этом сим­волической ссылке можно присвоить любое удобное имя, указать для од­ного файла несколько имен, что бывает удобно в ряде случаев. В про­грамме **Midnight Commander** перед их именами указывается символ «**@**». Кроме символических ссылок существуют *«жесткие» ссылки*, с помощью которых устанавливают связь между двумя или более файла­ми, имеющими разные имена, но одинаковое содержание. Любое изме­нение содержимого одного файла изменяет остальные. В какой-то мере жесткие ссылки можно рассматривать как резервирование файлов.

**Добавление пользователя**

Для регистрации нового пользователя в **Linux** администратор должен ввести следующую команду:

**useradd** имя\_нового\_пользователя

Имя пользователя (учетной записи) может содержать только символы ла­тинского алфавита и цифры. Не допускается использовать заглавные буквы. Правильный пример:

**useradd** egor

В результате выполнения команды **useradd** будет создан каталог /home/egor, а также подкаталоги для служебных файлов, которые предназначены для конфигурирования системы для данного пользователя. Кроме того, в обще­системных конфигурационных файлах будут добавлены записи, определяю­щие нового пользователя и его права.

Основная запись регистрации пользователя делается в текстовом файле /etc/passwd, в котором добавится, например, такая информация:

**egor:х:1001:1001::/home/egor:/bin/bash**

Но прежде чем новый пользователь войдет в систему, *администратор* дол­жен указать для него пароль, который пользователь в дальнейшем может самостоятельно изменить. Для задания пароля служит команда **passwd**:

> **passwd** egor

New UNIX passwrd: ввод\_пароля

**Return new UNIX password: повтор\_пароля**

passwd: all authentication tokens updated successfully

Для проверки того, что новый пользователь зарегистрирован в системе, можно, не прекращая текущей работы, войти в систему под другим именем и, соответственно, с другими правами. Для этого переключаемся на свобод­ную виртуальную консоль с помощью комбинации клавиш, например, **Alt+F3**. На экране третьей виртуальной консоли будет знакомое при­глашение:

**login:**

Если ввести правильно имя нового пользователя и пароль, то после входа в систему появится приглашение:

**[egor@localhost egor] $**

**Пользователи и группы**

Рассмотрим то, что определяют права пользовате­ля в системе: *идентификатор* *пользователя* и *идентификатор* *группы.*

Понятие ***идентификатор пользователя*** (**UID**) вводится для того, чтобы опе­рационная система могла пользоваться не текстовой строчкой, содержащей имя пользователя, а коротким цифровым кодом. Людям операционная сис­тема присваивает идентификаторы, начиная примерно с номера 100, но иногда с 500 или 1000. Вручную изменять идентификаторы не рекомендует­ся, для изменения привилегий пользователей существуют хорошие утилиты с графическим интерфейсом. Рассмотрим несколько строчек в файле /etc/passwd:

**egor:х:1001:1001:Egor:/home/egor:/bin/bash**

**root:x:0:0:root:/root:/bin/bash bin:x:1:1:bin:/bin:**

В файле /etc/passwd разделителем в строке служит символ «**:**», а конец запи­си (это одна строка) определяется символом перевода строки, который вво­дится нажатием клавиши **Enter**. Приведем назначение полей записи.

* ***имя пользователя***(*username, login*) — символьное имя пользователя для входа в систему, которое должно содержать не более 8 символов.
* ***пароль***(*password*) — пароль пользователя в закодированной форме.

|  |  |
| --- | --- |
| 256px-Information  256px-Information  256px-Information | Если в этом поле находятся символы «**х**», «**\***» или «**!!**», то используется система теневых паролей, когда закодированный пароль содержится в файле **/etc/shadow.** В этом файле для каждого пользователя также существует своя запись, состоящая из полей: 1 – имя пользователя, 2– зашифрованный пароль, 3 - количество дней с 1 января 1970г. до момента последней смены пароля и т.д. |

* ***идентификатор пользователя***(**UID**) — номер пользователя в системе.
* ***идентификатор группы***(**GID**) — идентификатор группы, в которую вхо­дит пользователь. Группы введены для удобства присвоения пользовате­лю и процессам прав на ресурсы системы, т. е. члены одной группы об­ладают одинаковыми правами на работы с файлами. Список групп находится в файле **/etc/group**.
* ***реальное имя пользователя***— имя пользователя, предназначенное для восприятия людьми.
* ***домашний каталог***— путь к домашнему каталогу пользователя.
* ***путь к исполняемому файлу командной оболочки***— при входе в систему запускается указанная в этом поле оболочка, например **/bin/bash**.

|  |  |
| --- | --- |
| 256px-Information  256px-Information  256px-Information | Администратор может зарегистрировать нового пользователя или изменить права уже существующего, вручную откорректировав строки в файле **/etc/passwd**, но в этом случае придется проделать много дополнительной ра­боты по определению прав пользователя и созданию его домашнего каталога и конфигурационных файлов. |

**Атрибуты файлов**

Каждый файл и каталог в операционной системе **Linux** имеет ярлык (*эти­кетку, атрибуты, attributes*), которая определяет права доступа к данно­му файлу тем или иным пользователям. Это похоже на атрибуты файла в **Windows**, но там всего четыре атрибута, которые определяют видимость файла (*архивный, системный, только для чтения, скрытый*), мало влияющих на текущую работу. В **Linux** с атрибутами файла работает непосредственно ядро, которое по ним определяет, что может делать с данным файлом кон­кретный пользователь: читать, изменять или запускать файл (программу) на исполнение.

В **Linux** права доступа к файлу определены для трех уровней пользователей:

* для ***владельца файла***, который создал этот файл;
* для ***пользователей***, принад­лежащих к группе, например, группа сотрудников, работающая над одним проектом;
* для ***всех прочих пользователей***.

И только администратор имеет право проводить любые операции с файлами.

Пользователь для входа в систему регистрируется, вводя свой пароль и имя. После этого он получает доступ к файловой системе. Для ускорения быст­родействия системы используются идентификаторы пользователя **UID** и группы **GID**. Соответственно у атрибутов файлов имеется шаблон, кото­рый определяет принадлежность файла конкретному пользователю и группе. Если пользователь не является владельцем файла и не входит в состав груп­пы, то он относится к *прочим* (others), которые имеют очень мало прав, а то и вовсе ими не обладают.

Учитывая такой принцип деления пользователей, для описания прав доступа к файлам используется шаблон: **Владелец Группа Прочие**

Во многих случаях для облечения понимания того, как распределяются права, используется запись атрибутов файла в виде символов «***x***», «***w***» и «***r***». Связь такого представления с цифровыми системами счисле­ния показана таблице:

Таблица 1. Связь записи атрибутов файла с цифровыми системами счисле­ния

|  |  |  |
| --- | --- | --- |
| Двоичная | Восьмеричная | Права доступа |
| 000 | 0 | **---** |
| 001 | 1 | **--x** |
| 010 | 2 | **-w-** |
| 011 | 3 | **-wx** |
| 100 | 4 | **r--** |
| 101 | 5 | **r-x** |
| 110 | 6 | **rw-** |
| 111 | 7 | **rwx** |

Такое символьное представле­ние атрибутов файлов произошло от восьмеричной системы счисления (*традиционно в* ***Unix****-системах используется восьмеричная система счисления для записи чисел*), а также от того, что для идентификации каждого уровня пользователей ис­пользуется три бита (тремя битами можно обозначить только восемь значе­ний). В итоге, получилось следующее:

* право исполнения (**--x**), сокр. от *execute* — исполнять;
* право редактирования (**-w-**), сокр. от *write* — писать;
* право чтения (**r--**), сокр. от *read* — читать.

Теперь рассмотрим примеры различных прав на файл.

Если существует файл, с которым любой пользователь может проводить лю­бые операции, то он будет иметь следующие атрибуты:

**rwxrwxrwx** или **777**

Для файлов данных, которые не являются программами и скриптами и ко­торые могут читать и перезаписывать любые пользователи, используется следующий вариант (разрешено чтение и редактирование для всех):

**rw-rw-rw-** или **666**

Когда владельцу файла необходимо сохранить конфиденциальную инфор­мацию, например это личное письмо, то следует установить следующие ат­рибуты:

**rw-** или **600**

Все, что было сказано об атрибутах файлов, касается и каталогов, но смысл прав несколько иной. Если у пользователя есть право на чтение (***r***) катало­га, то это означает, что в данном каталоге он может вывести на экран со­держимое этого каталога. Чтобы записывать и удалять в каталоге файлы, пользователь должен иметь право записи (***w***). Право на исполнение (***х***) для каталога означает, что пользователь может войти в ката­лог и просмотреть его содержание.

При просмотре файловой системы **Linux** в программе **Midnight Commander** в режиме расширенного формата отображения, можно заметить, что рас­смотренные ранее атрибуты файлов предваряются еще одним символом, т. е. отображается 10 символов, а не 9. «Лишний» символ, который предва­ряет атрибуты прав доступа, обозначает тип файла:

**-** — обычный файл; **d** — каталог; **l** — символическая ссылка;

**b** — файл блочного устройства; **с** — файл символьного устройства;

**s** — доменное гнездо; **р** — именованный канал

Также в атрибуты файла входят поля, в которых указывается время послед­него доступа к файлу, время модификации и время изменения атрибутов.

Кроме этого, каждый файл имеет два идентификатора, определяющего его принадлежность – владелец файла и группа пользователей:

-rw-r--r-- 1 egor users 6043 2008-01-11 01:28 myfile

Имя

Время создания файла

Группа

Владелец

Права доступа

**Дополнительные флаги**

После рассмотрения прав на файлы возникает вопрос, а как рядовому поль­зователю запускать системные программы, которые принадлежат админист­ратору, ведь он не владелец этих файлов. Для таких целей существует меха­низм ограничения прав пользователя, когда он работает с системными файлами. Например, при изменении собственного пароля пользователю по­зволено читать информацию из файла паролей, но изменить в нем он может только ту информацию, которая относится к собственному паролю. Реали­зация такого способа доступа к файлам производится с помощью флага Set-UID (**SUID**).

Флаг **SUID** обозначается символом «***s***» на позиции права исполнения (**--s- -х- -х**). В тех случаях, когда нет права исполнения для самого владельца файла, в атри­бутах файла появится символ «***s***».

В восьмеричной систе­ме счисления для указания флага **SUID** используется число **4000**, например, атрибут **--s- -х- -х-** в восьме­ричном виде будет иметь вид **4111**.

Для групп аналогичный флаг носит название Set-GID (**SGID**) и указывается в зоне определения прав группы. Восьмеричное значение флага **SGID** равно **2000**.

Флаг в зоне прав прочих пользователей называется Sticky-Bit (*sticky*, клей­кий), используется для запрещения удаления кода программы из памяти даже тогда, когда она уже выполнена. В операционной системе **Linux** особой нужды в нем нет, поэтому его используют только для ряда каталогов, в ча­стности для запрета на удаление временных файлов в каталоге для всех, кроме владельца этих файлов. Обозначается флаг Sticky-Bit символом «***t***», например, **rwxrwxrwt** или **1777**.

**Изменение прав пользователей**

***Для изменения права доступа к каждого файлу или каталогу*** используется команда **chmod**. Параметры прав доступа можно задавать как в восьмеричной форме, так и используя символьную нотацию. С этой командой использу­ются следующие ключи:

**chmod** Права Имя\_файла

При восьмеричной форме задания прав доступа команда **chmod** устанавлива­ет права доступа, как это указано в параметрах:

> **ls -l my\_file**

-rw-rw-r- 1 my my 10 Apr my\_file

> **chmod 600 my\_file**

> **ls -l my\_file**

-rw 1 my my 10 Apr my\_file

***Для смены права собственности на файл или ката­лог*** существует команда **chown**. Допускается изменять как владельца, так и группу. Следует отметить, что если пользователя или группы не существует, владелец файла не меня­ется, для группы в таком случае устанавливается группа по умолчанию.

**chown** Владелец[:Группа] Имя\_файла

Например, изменяем владельца файла myfile

**chown** ivan myfile

Теперь изменяем группу для файла myfile

**chown** :users myfile

А теперь изменяем одновременно владельца и группу для файла myfile

**chown** ivan:users myfile

***Для изменения групповых прав собственности*** имеется команда **chgrp**, ко­торая, правда, не обладает такой гибкостью, как команда **chown**:

**chgrp** Группа Имя\_файла

***Для*** ***выяснения к какой группе вы принадлежите***:

> **groups:**

groups root

Для уточнения того, какие пользователи принадлежат группе, полезно по­интересоваться содержимым файла **/etc/group** или использовать команду groups с именем пользователя:

> **groups** my

my : my

**Примеры рассмотренных команд**

**cd ~** переходим в свой домашний каталог

**touch myfile** создаем новый пустой файл myfile, для этого требуются права пользователя *root* иначе появляется уведомление в отказанном доступе:

touch: невозможно выполнить touch для `myfile': Permission denied

Входим под пользователем *root* и повторяем.

su

touch myfile

**ls -l** смотрим, что есть сейчас в домашнем каталоге

…

-rw-r--r-- 1 egor users 6043 2008-01-11 01:28 .muttrc

-rw-r--r-- 1 root root 0 2008-03-18 21:14 myfile

drwx------ 3 egor users 96 2008-03-18 21:12 .ooo-2.0-pre

…

**chmod 740 myfile**  изменяем права у файла **myfile**

**ls -l**  убеждаемся, что все сделано

…

-rw-r--r-- 1 egor users 6043 2008-01-11 01:28 .muttrc

-rwxr----- 1 root root 0 2008-03-18 21:14 myfile

drwx------ 3 egor users 96 2008-03-18 21:12 .ooo-2.0-pre

…

**groups root** смотрим к какой группе принадлежит пользователь *root*

root : root

**touch sales** создаем новый пустой файл **sales**

**ln -s ~/sales ~/Desktop/saleslink** создаем на рабочем столе

символьную ссылку на него

**cd ~/Desktop** переходим на рабочий стол

**ls -l** смотрим, что тут есть

…

-rw-r--r-- 1 root root 642 2008-02-25 13:02 Office.desktop

lrwxrwxrwx 1 root root 11 2008-03-18 21:15 saleslink -> /root/sales

-rw-r--r-- 1 root root 4795 2008-02-25 13:02 trash.desktop

…

**Практические задания**

1. Создайте нового пользователя. Задайте ему пароль. Убедитесь, что создана его домашняя папка. Покажите, в какую группу он входит.
2. Докажите записью в служебных файлах, что добавили нового пользователя. Расшифруйте эту запись.
3. Войдите в систему под новым пользователем.
4. Покажите, к какой группе относится новый пользователь.
5. Создайте файл и символьную ссылку на него.
6. Определите, сколько символьных ссылок в папке **/usr** и на что они ссылаются.
7. Переведите из восьмеричной формы записи прав доступа в символьную 641. Переведите в восьмеричную форму записи прав доступа из символьной rw-r------.
8. Определите, кто является владельцем и домашнего каталога пользователя *student* икакие права на него установлены.
9. Создайте новый файл. Смените его владельца и группу пользователей. Установите бит **SUID** для этого файла.
10. Выясните, кто еще из пользователей принадлежит группе users