

ЛАБОРАТОРНАЯ РАБОТА № 4

РАБОТА С ФАЙЛАМИ В КОМАНДНОЙ СТРОКЕ

Цель работы: Изучение основных команд для работы с файлами и каталогами. Приобретение навыков работы с командной строкой в графическом и консольном интерфейсах POSIX систем.

Все консольные команды читают информацию со стандартного потока ввода и выводят ее в стандартный поток вывода. По умолчанию стандартным потоком вывода является текущая консоль, а стандартным потоком ввода - клавиатура. Стандартный поток может быть перенаправлен в любой файл, в том числе и в файл любого устройства из каталога устройств `/dev`. Символом перенаправления выходного потока служит значок `>`, а входного - `<`. Например, команда `ls > work/lab5.txt` создает новый или заменяет старый файл `lab5.txt` в каталоге `work` и записывает в него результат выполнения команды `ls`.

Для того, чтобы выходной поток добавлялся к старому содержимому файла, используется символ перенаправления `>>`. Так, если после предыдущей команды выполнить команду `cat /var/log/dmesg >> work/lab4.txt`, то в файл `work/lab5.txt` будет добавлен протокол загрузки ядра.

Команды могут объединяться в цепочки с помощью символа `|`. При объединении выходной поток предыдущей команды поступает во входной поток следующей.

Например, цепочка команд

```
cat /var/log/dmesg | head 10 | less S
```

интерпретируется следующим образом: получить протокол загрузки ядра, выделить в нем первые 10 строк и послать их в программу просмотра в интерактивном режиме.

Права на файлы

Для разграничения прав доступа к файлам и каталогам каждому объекту в файловой системе Linux присваивается маска прав доступа и определяется владелец объекта и группа владельцев объекта. Владелец объекта может быть любой из зарегистрированных в системе пользователей.

Группой-владельцем может быть любая из зарегистрированных в системе групп пользователей. Маска прав доступа определяет - какие права имеют отдельные пользователи на манипуляции с объектом.

Она включает следующие основные поля:

- поле разрешения чтения для владельца файла (UR) – если установлено, то владелец имеет право читать файл, для каталогов чтение обозначает право на доступ к файлам в каталоге.
- поле разрешения записи для владельца файла (UW) – если установлено, то владелец имеет право изменять файл. Для каталогов право записи разрешает создание и удаление файлов в каталоге.
- поле разрешения исполнения для владельца файла (UX) – если установлено, то владелец имеет право исполнять файл. Для каталогов исполнение обозначает право на просмотр содержимого каталога.
- Аналогичные три поля для членов группы-владельца (GR, GW, GX) определяют права, которые будут иметь на доступ к объекту пользователи, входящие в группу владельца.
- Те же три поля задаются для всех остальных пользователей (OR, OW, OX), определяя их права на доступ к файлу.
- Поле установки идентификатора пользователя при исполнении (SUID) – если установлено, то при исполнении файла он будет выполняться от имени пользователя-владельца файла. Установка данного поля для каталога указывает, что владельцем всех вновь создаваемых в каталоге файлов будет пользователь-владелец каталога. Данный флаг позволяет создавать программы, которые имеют права на доступ к объектам операционной системы, даже если у пользователя их запустившего таких прав нет.
- Поле установки идентификатора группы (SGID) аналогично полю установки идентификатора пользователя, но изменяет при исполнении не пользователя, а группу.

Полный список возможных атрибутов доступа можно получить в справочном руководстве.

Маска прав на доступ представляется в виде двоичного числа, в котором 1 в соответствующей позиции обозначено наличие права. Формат маски следующий:

s	S	t	r	w	x	r	w	x	r	w	x
SUID	SGID	см. рук-во	UR	UW	UX	GR	GW	GX	OR	OW	OX

Маску часто записывают в символьном виде в порядке следования битов, указывая соответствующий символ, если бит установлен, или символ -, если он сброшен. Старшие сброшенные биты при этом обычно не указываются.

Например, маска `rwX-r-X-` обозначает, что владелец имеет полные права на объект, члены группы и все остальные – только права на чтение и исполнение файла. Также маска прав может быть записана в восьмеричном виде. При этом та же маска будет иметь вид `0755`.

Маску часто записывают в символьном виде в порядке следования битов, указывая соответствующий символ, если бит установлен, или символ -, если он сброшен.

Старшие сброшенные биты при этом обычно не указываются.

Например, маска `rwX r-X-` обозначает, что владелец имеет полные права на объект, члены группы и все остальные – только права на чтение и исполнение файла. Также маска прав может быть записана в восьмеричном виде. При этом та же маска будет иметь вид `0755`.

Как мы уже знаем, в системах UNIX для выдачи содержимого каталога используется

команда `ls`. Ее формат следующий:

`ls [опции] [<путь>][<имя файла>]`

где `<путь>` - указание имени каталога, листинг которого следует произвести.

`<имя файла>` - имя файла (или шаблон группы файлов), список которого хотелось бы просмотреть.

Если не указан путь, то подразумевается текущий каталог.

Если не указано имя файла (или шаблон), подразумеваются все файлы каталога.

Опции:

`l` - листинг файлов в полном формате.

`F` - имена файлов и каталогов различаются.

`a` - вывод в листинге всех файлов и каталогов, а не только тех, к которым есть доступ.

`d` - вывод имени каталога;

`i` — вывод номера описателя файла (`i-node`).

Возможны и другие опции. Чтобы узнать их, смотри описание своей версии UNIX или выполните команду

`man ls`

Комбинация опций:

`al` - вывод в листинге всех файлов в полном формате.

`ld` - просмотр атрибутов (полный листинг) каталогов.

Пример:

`ls ld thank you` - просмотр с атрибутами и правами доступа директорий или

файлов с именами `thank` и `you`.

`ls l myfile` выдает атрибуты и права доступа файла `myfile`.

Как ясно из этого описания, команда `ls` не выдает полного имени текущего каталога.

Это делается командой `pwd`. Команда `pwd` выводит в стандартный поток вывода

полное имя текущего каталога.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Для выполнения данной лабораторной работы запустите операционную систему Linux, установленную на виртуальную машину в первой лабораторной работе.

1. Изучите файлы в своем домашнем каталоге. Перейти в него можно по команде `cd ~` хотя по умолчанию после входа в систему пользователь будет находиться в

своем домашнем каталоге. Напомним, что посмотреть содержимое каталога можно с помощью команды `ls`, причем при использовании ключа `l` мы получаем вывод содержимого в так называемом длинном формате, а ключ `a` позволяет увидеть и скрытые файлы тоже.

2. Создайте в своем домашнем каталоге новый каталог `Files`.

3. Создайте в этом каталоге файлы `NewFile` и в качестве данных впишите в него свою фамилию и имя. Выведите на экран содержимое файла.

4. С помощью команды `ln` создайте ещё один файл (`NewName`), который будет ссылаться на файл `NewFile`. С помощью команды `ls -il ~/Files` получите информацию о созданных файлах. Убедитесь, что файлы `NewFile` `NewName` ссылаются на один и тот же `inode`, а значит содержат одну и ту же информацию.

13. Перейдите в свой домашний каталог и создайте там ещё одну жесткую ссылку

`SecondName` на тот же файл `NewName`.

4. В отличие от обычных жестких ссылок (Вы только что создали такие ссылки) мягкие ссылки могут ссылаться на каталоги и на файлы, расположенные на других разделах или носителях. В качестве значений, содержащихся в таких файлах (в мягких ссылках), будут пути к файлам или каталогам. Другими словами, это похоже на косвенную

адресацию: мы в таких файлах храним не данные, а адреса данных.

Находясь в своём домашнем каталоге, с помощью команды `ln -s` создайте мягкую ссылку на файл `NewName`. Изучите inode всех созданных Вами файлов (с помощью команды `ls -li`).