

## ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4. «ИЗУЧЕНИЕ СТРУКТУРЫ ФАЙЛОВОЙ СИСТЕМЫ ОС LINUX»

### Теоретическая часть

В ходе выполнения данной работы будут изучены структуры файловой системы ОС LINUX, команды создания, удаления, модификации файлов и каталогов, функции манипулирования данными.

#### Файловая структура системы LINUX

В операционной системе LINUX файлами считаются обычные файлы, каталоги, а также специальные файлы, соответствующие периферийным устройствам (каждое устройство представляется в виде файла). Доступ ко всем видам файлам однотипный. Такой подход обеспечивает независимость программы пользователя от особенностей ввода/вывода на конкретное внешнее устройство.

Файловая структура LINUX имеет иерархическую древовидную структуру. В корневом каталоге размещаются другие каталоги и файлы, включая 5 основных каталогов:

bin - большинство выполняемых командных программ и shell-процедур;

tmp - временные файлы;

usr - каталоги пользователей;

etc - административные утилиты и файлы;

dev - специальные файлы, представляющие периферийные устройства.

Текущий каталог - это каталог, в котором в данный момент находится пользователь. При наличии прав доступа пользователь может перейти в любой каталог.

Текущий каталог обозначается точкой (.), родительский каталог, которому принадлежит текущий, обозначается двумя точками (..).

Полное имя файла может включать имена каталогов, включая корневой, разделенных косой чертой, например: /home/student/file.txt. Первая косая черта обозначает корневой каталог, и поиск файла будет начинаться с него, а затем в каталоге home, затем в каталоге student.

Один файл можно сделать принадлежащим нескольким каталогам. Для этого используется команда **ln (link)**:

### **ln <имя 1-го файла> <имя 2-го файла>**

Имя 1-го файла - это полное имя файла, с которым устанавливается связь; имя 2-го файла - это полное имя файла в новом каталоге, где будет использоваться эта связь. Новое имя может не отличаться от старого. Каждый файл может иметь несколько связей, то есть он может использоваться в разных каталогах под разными именами. Команда **ln** с аргументом **-s** создает символическую связь:

### **ln -s <имя 1-го файла> <имя 2-го файла>**

Здесь имя 2-го файла является именем символической связи. Символическая связь является особым видом файла, в котором хранится имя файла, на который эта связь ссылается. LINUX работает с символической связью не так, как с обычным файлом - например, при выводе на экран содержимого символической связи появятся данные файла, на который эта связь ссылается.

В LINUX различаются 3 уровня доступа к файлам и каталогам:

- 1) доступ владельца файла;
- 2) доступ группы пользователей, к которой принадлежит владелец файла;
- 3) остальные пользователи.

Для каждого уровня существуют свои байты атрибутов, значение которых расшифровывается следующим образом:

- r – разрешение на чтение;
- w – разрешение на запись;
- x – разрешение на выполнение;
- – отсутствие разрешения.

Первый символ байта атрибутов определяет тип файла и может принимать следующие значения:

- – обычный файл;
- d – каталог;
- l – символическая связь;

в – блок-ориентированный файл, который соответствует накопителям на магнитных дисках;

с – байт-ориентированный файл, который соответствует принтеру или терминалу.

В домашнем каталоге пользователь имеет полный доступ к файлам (READ, WRITE, EXECUTE; r, w, x).

Атрибуты файла можно просмотреть командой `ls -l` и они представляются в следующем формате:

```
d rwx rwx rwx
| | | |
```

```

| | | | Доступ для остальных пользователей
| | | | Доступ к файлу для членов группы
| | | | Доступ к файлу владельца
| | | | Тип файла (директория)
    
```

Пример. Командой **ls -l** получим листинг содержимого текущей директории student:

```

- rwx --- --- 2 student 100 Mar 10 10:30 file_1
- rwx --- r-- 1 adm 200 May 20 11:15 file_2
- rwx --- r-- 1 student 100 May 20 12:50 file_3
    
```

После байтов атрибутов на экран выводится следующая информация о файле:

- число связей файла;
- имя владельца файла;
- размер файла в байтах;
- дата создания файла (или модификации);
- время;
- имя файла.

Атрибуты файла и доступ к нему, можно изменить командой **chmod <коды защиты> <имя файла>**.

Коды защиты могут быть заданы в числовом или символьном виде. Для символьного кода используются:

- (+) - добавить права доступа;
- (-) - отменить права доступа;
- r,w,x - доступ на чтение, запись, выполнение;
- u,g,o - владельца, группы, остальных.

Коды защиты в числовом виде могут быть заданы в восьмеричной форме. Для контроля установленного доступа к своему файлу после каждого изменения кода защиты нужно проверять свои действия с помощью команды **ls -l**.

Примеры:

**chmod g+rw,o+r <файл>** - установка атрибутов чтения и записи для группы и чтения для всех остальных пользователей;

**ls -l <файл>** - чтение атрибутов файла;

**chmod o-w <файл>** - отмена атрибута записи у остальных пользователей;

**>letter** - создание файла letter. Символ > используется как для переадресации, так и для создания файла;

**cat** - вывод содержимого файла;

**cat <файл1> <файл2> > <файл3>** - конкатенация файлов (объединение);

**mv <старое\_имя> <новое\_имя>** - переименование файла;

**mv <файл> <directory>** - перемещение файла в указанную директорию;

**rm <файл>** - удаление файла;

**cp <файл1> <файл2>** - копирование файла с переименованием;

**mkdir <каталог>** - создание каталога;

**rm <каталог>** - удаление каталога;

**ls [acdfgilqrstv CFR] <каталог|файл>** - вывод содержимого. Значения аргументов:

- l – вывести всю информацию о файле;
- t – сортировка по времени модификации файлов;
- a – в список включаются все файлы, в том числе и те, которые начинаются с точки;

- s – размеры файлов указываются в блоках;

- d – вывести имя самого каталога, но не содержимое;

- r – сортировка строк вывода;

- i – указать идентификационный номер каждого файла;

- v – сортировка файлов по времени последнего доступа;

- q – непечатаемые символы заменить на знак ?;

- c – использовать время создания файла при сортировке;

- g – то же что -l, но с указанием имени группы пользователей;

- f – вывод содержимого всех указанных каталогов, отменяет флаги -l, -t, -s, -r и активизирует флаг -a;

- c – вывод элементов каталога в несколько столбцов;

- F – добавление к имени каталога символа / и символа \* к имени файла, для которых разрешено выполнение;

- R – рекурсивный вывод содержимого подкаталогов заданного каталога.

**cd <каталог>** - переход в каталог. Если параметры не указаны, то происходит переход в домашний каталог пользователя.

**pwd** - вывод имени текущего каталога;

**grep [-vcilns] [шаблон поиска] <файл>** - поиск файлов с указанием или без указания шаблона.

Значение ключей:

- v – выводятся строки, не содержащие шаблон поиска;

- c – выводится число строк, содержащих или не содержащих шаблон;

- i – регистронезависимый поиск;

- l – выводятся только имена файлов, содержащие указанный шаблон;

- n – перенумерация выводимых строк;
- s – формируется только код завершения.

Примеры.

1. Напечатать имена всех файлов текущего каталога, содержащих последовательность "student" и имеющих расширение .txt:

**grep -l student \*.txt.**

2. Определить имя пользователя, входящего в ОС LINUX с терминала tty23:

**who | grep tty23.**

### Порядок выполнения работы

1. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами.
2. Используя команды ОС LINUX, создать два текстовых файла.
3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
4. Создать новую директорию и переместить в нее полученные файлы.
5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа.
6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
7. Просмотреть атрибуты файлов.
8. Создать еще один каталог.
9. Установить дополнительную связь объединенного файла с новым каталогом, но под другим именем.
10. Создать символическую связь.
11. Сделать текущим новый каталог и вывести на экран расширенный список информации о его файлах.
12. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов.
13. Получить информацию об активных процессах и имена других пользователей.
14. Сдать отчет о работе и удалить свои файлы и каталоги.
15. Выйти из системы.

- n – перенумерация выводимых строк;
- s – формируется только код завершения.

Примеры.

1. Напечатать имена всех файлов текущего каталога, содержащих последовательность "student" и имеющих расширение .txt:

```
grep -l student *.txt.
```

2. Определить имя пользователя, входящего в ОС LINUX с терминала tty23:

```
who | grep tty23.
```

### Порядок выполнения работы

1. Ознакомиться с файловой структурой ОС LINUX. Изучить команды работы с файлами.
2. Используя команды ОС LINUX, создать два текстовых файла.
3. Полученные файлы объединить в один файл и его содержимое просмотреть на экране.
4. Создать новую директорию и переместить в нее полученные файлы.
5. Вывести полную информацию обо всех файлах и проанализировать уровни доступа.
6. Добавить для всех трех файлов право выполнения членам группы и остальным пользователям.
7. Просмотреть атрибуты файлов.
8. Создать еще один каталог.
9. Установить дополнительную связь объединенного файла с новым каталогом, но под другим именем.
10. Создать символическую связь.
11. Сделать текущим новый каталог и вывести на экран расширенный список информации о его файлах.
12. Произвести поиск заданной последовательности символов в файлах текущей директории и получить перечень соответствующих файлов.
13. Получить информацию об активных процессах и имена других пользователей.
14. Сдать отчет о работе и удалить свои файлы и каталоги.
15. Выйти из системы.

### Контрольные вопросы

1. Что считается файлами в ОС LINUX?
2. Объясните назначение связей с файлами и способы их создания.
3. Что определяет атрибуты файлов и каким образом их можно просмотреть и изменить?
4. Какие методы создания и удаления файлов, каталогов Вы знаете?
5. В чем заключается поиск по шаблону?
6. Какой командой можно получить список работающих пользователей и сохранить его в файле?

### Список использованных источников

1. Колисниченко Д. Н. Linux. От новичка к профессионалу. - СПб.: БХВ-Петербург, 2010. - 780 с.