Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного образовательного
учреждения высшего образования
**«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»**
(КФ МГТУ им. Н.Э. Баумана)

Ю.С. Белов, Е.А. Черепков

ЗНАКОМСТВО С ФАЙЛОВОЙ СИСТЕМОЙ LINUX

Методические указания к выполнению лабораторной работы

по курсу «Операционные системы»

Калуга – 2018

УДК 004.62
ББК 32.972.1
Б435

Методические указания составлены в соответствии с учебным планом КФ МГТУ им. Н.Э. Баумана по направлению подготовки 09.03.04 «Программная инженерия» кафедры «Программного обеспечения ЭВМ, информационных технологий».

Методические указания рассмотрены и одобрены:

- Кафедрой «Программного обеспечения ЭВМ, информационных технологий» (ИУ4-КФ) протокол № 3 от «24» октября 2018 г.

Зав. кафедрой ИУ4-КФ _____ к.т.н., доцент Ю.Е. Гагарин

- Методической комиссией факультета ИУ-КФ протокол № 3 от «29» октября 2018 г.

Председатель методической
комиссии факультета ИУ-КФ _____ к.т.н., доцент М.Ю. Адкин

- Методической комиссией
КФ МГТУ им.Н.Э. Баумана протокол № 2 от «6» ноября 2018 г.

Председатель методической комиссии
КФ МГТУ им.Н.Э. Баумана
_____ д.э.н., профессор О.Л. Перерва

Рецензент:
к.т.н., доцент кафедры ИУ3-КФ _____ А.В. Финошин

Авторы
к.ф.-м.н., доцент кафедры ИУ4-КФ _____ Ю.С. Белов
ассистент кафедры ИУ4-КФ _____ Е.А. Черепков

Аннотация

Методические указания к выполнению лабораторной работы по курсу «Операционные системы» содержат общие сведения о работе с файлами и каталогами в ОС Linux Slackware, а также синтаксис основных команд для работы с файловой системой.

Предназначены для студентов 3-го курса бакалавриата КФ МГТУ им. Н.Э. Баумана, обучающихся по направлению подготовки 09.03.04 «Программная инженерия».

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Операционные системы» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета «Информатика и управление» Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов 3-го курса направления подготовки 09.03.04 «Программная инженерия», содержат краткое описание команд для работы с файлами и каталогами.

Методические указания составлены для ознакомления студентов с работой с каталогами и файлами в операционной системе Linux. Для выполнения лабораторной работы студенту необходимы минимальные знания об операционной системе Linux.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является приобретение практических навыков по работе с файлами и каталогами в ОС Linux.

Основными задачами выполнения лабораторной работы являются:

1. Ознакомиться с устройством файловой системы ОС Linux.
2. Получить навыки работы с файлами и каталогами ОС Linux.
3. Изучить основные команды работы с файлами и каталогами ОС Linux.

Результатами работы являются:

1. Демонстрация выполнения команд для работы с файлами и каталогами.
2. Подготовленный отчет.

КРАТКАЯ ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

Все файлы в ОС Linux имеют один и тот же формат - байтовый поток. Байтовый поток представляет собой просто последовательность байтов. Это позволяет распространять понятие файла в системе Linux на все компоненты данных. Каталоги и устройства классифицируются как файлы. Рассматривая все эти объекты как файлы, Linux позволяет упростить организацию данных и обмен ими. Данные, записанные в файле, можно посылать непосредственно на устройство, например на экран, потому что устройство сопрягается с системой, используя тот же формат байтового потока, который применяется в обычных файлах.

Этот же формат файлов используется при создании других компонентов операционной системы. Интерфейс, обеспечивающий доступ к какому-либо устройству, например к экрану или клавиатуре, проектируется как файл. Другие компоненты, в частности каталоги, сами по себе являются файлами типа потоков байтов, но имеют особую внутреннюю организацию. Файл каталога содержит информацию о каталоге, оформленную в специальном формате каталога. Поскольку все эти различные компоненты рассматриваются как файлы, можно сказать, что они представляют собой различные типы файлов. Устройство посимвольного ввода-вывода — один тип файла. Каталог — другой тип файла. Число типов файлов зависит от конкретной реализации ОС Linux, однако существуют пять стандартных типов: обычные файлы, файлы каталогов, каналы, действующие по принципу очереди, файлы устройств посимвольного ввода-вывода и файлы устройств поблочного ввода-вывода.

Файловая структура

Файлы в операционной системе Linux организованы в иерархическую систему каталогов. Каталог может содержать файлы и другие каталоги. В этом смысле каталоги выполняют две важные функции. Во-первых, в каталоге хранятся файлы, подобно папкам в

ящике картотеки, а во-вторых, каталог соединяется с другими каталогами, как ветвь дерева соединяется с другими ветвями. По отношению к файлам каталоги выполняют роль ящиков картотеки, в каждом из которых хранятся несколько папок. Для того чтобы взять одну из них, нужно открыть ящик. Следует отметить, однако, что, в отличие от ящиков картотеки, каталоги могут содержать не только файлы, но и другие каталоги. Именно таким образом каталог может соединяться с другим каталогом.

Из-за сходства с деревом такую структуру часто называют древовидной структурой, данную структуру можно назвать структурой «с родительско-дочерними связями». Аналогичным образом любой каталог является подкаталогом другого каталога. Каждый каталог может содержать множество подкаталогов, но сам должен быть потомком только одного родительского каталога.

Файловая структура ОС Linux разветвляется на несколько каталогов, начиная с корневого, /. В корневом каталоге имеется несколько системных каталогов, которые включают файлы и программы, относящиеся к самой ОС Linux. Корневой каталог, кроме того, содержит каталог home, в котором могут храниться начальные каталоги всех пользователей системы. Начальный каталог каждого пользователя, в свою очередь, будет включать в себя каталоги, которые пользователь создает для своих нужд. Каждый из этих каталогов тоже может содержать каталоги. Все эти вложенные каталоги ответвляются от начального каталога пользователя, как показано на рис. 1.

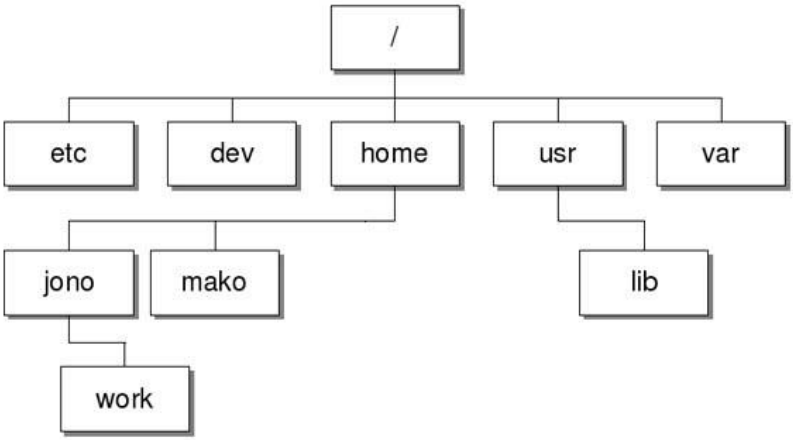

Рис.1. Файловая структура Linux

Начальные каталоги

Зарегистрировавшись в системе, вы попадаете в свой начальный каталог. Имя, присвоенное этому каталогу системой, совпадает с вашим регистрационным именем. Все файлы, создаваемые для нового пользователя, помещаются в его начальный каталог. В этом каталоге можно создавать подкаталоги и размещать в них файлы. Создавать свои начальные каталоги могут и другие пользователи системы. Имя начального каталога каждого пользователя совпадает с регистрационным именем этого пользователя.

Если при проведении какой-либо операции над файлами имя каталога не указывается, то используется каталог по умолчанию, который называют также рабочим каталогом.

При регистрации в системе в качестве рабочего принимается ваш начальный каталог, имя которого обычно совпадает с вашим регистрационным именем. Рабочий каталог можно сменить с помощью команды cd. В процессе смены рабочего каталога вы переходите из одного каталога в другой.

Системные каталоги

Корневой каталог, являющийся началом файловой структуры ОС Linux, содержит ряд системных каталогов. В них хранятся файлы и программы, служащие для управления системой и ее сопровождения. Многие из этих каталогов включают подкаталоги с программами, предназначенными для выполнения конкретных задач. Основные системные каталоги перечислены в табл. 1.

Таблица 1. Стандартные системные каталоги в ОС Linux

/	Служит начальной единицей файловой системы и называется *корнгеым*
/home	Содержит *начальные* каталоги пользователей
/bin	Содержит все стандартные команды и утилиты
/usr	Содержит файлы и команды, используемые системой; разбит на несколько подкаталогов
/usr/bim	Содержит команды и утилиты, применяемые пользователем
/usr/sbin	Содержит команды системного администрирования
/usr/lib	Содержит библиотеки языков программирования
/usr/doc	Содержит документацию ОС Linux
/usr/man	Содержит файлы справочных руководств
/usr/spool	Содержит буферные файлы (например, создаваемые для заданий по сети)
/sbin	Содержит команды администрирования системы для начальной загрузки
/var	Содержит изменяющиеся файлы, например файлы почтовых ящиков
/dev	Содержит файлы, обеспечивающие взаимодействие пользователя с устройствами на пример с терминалами и принтерами
/etc	Содержит файлы конфигурации системы и прочие системные файлы

ИМЕНОВАНИЕ ФАЙЛОВ И КАТАЛОГОВ

Linux образует уровень абстракции. Текстовая информация, подключенные устройства, запущенные процессы представлены в системе как файлы. Например, выполнение команды «cat /proc/loadavg» выведет статистику использования ресурсов процессора. Таким образом, через чтение файла были получены реальные характеристики системы на текущий момент времени.

Именование файлов и каталогов в Linux имеет ряд правил не зависимо от того какую информацию они содержат. Активным пользователям других операционных систем нужно обратить на это особое внимание. Может найтись несколько отличий от привычной схемы.

Правила именования файлов и каталогов:

- Имя должно быть не длиннее 255 символов.
- Имена чувствительны к регистру. Например, «example.txt» и «eXaMpLe.txt» являются разными файлами, даже если расположены в одном каталоге.
- Допускается использование символов нижнего подчеркивания, точек и дефисов.
- Если имя файла или каталога начинается с точки, он будет воспринят системой как скрытый.
- В имени также могут содержаться пробелы. Если консольная команда требует указания имени файла или директории, в котором содержатся пробелы, его следует заключить в кавычки. Например, «ls "my directory/test file.xlsx"».
- Допускается использование, как латиницы, так и раскладки национального языка. Linux полностью поддерживает кодировку UTF-8.

Путевые имена

Имя, которое дается каталогу или файлу при его создании, не является полным. Полным именем каталога является его путевое имя.

Иерархические связи, существующие между каталогами, образуют пути, и эти пути можно использовать для однозначного указания каталога или файла и обращения к нему. На рис, 2 показан путь от корневого каталога, /, через каталог home в каталог robert, и путь от корневого каталога через каталоги home и chris в каталог reports. Путевое имя файла weather (рис. 3) состоит из имен корневого каталога, каталогов home и chris и имени файла: /home/chris/weather (корневой каталог обозначен первой косой чертой).

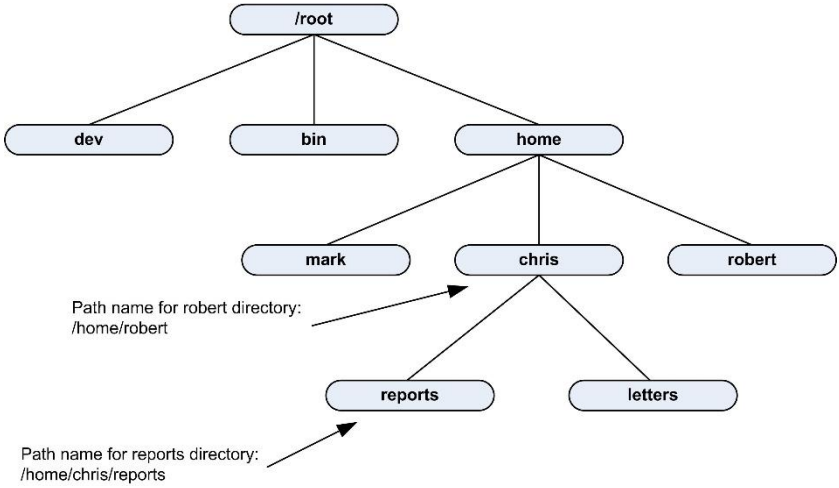

Рис.2. Путевые имена каталогов

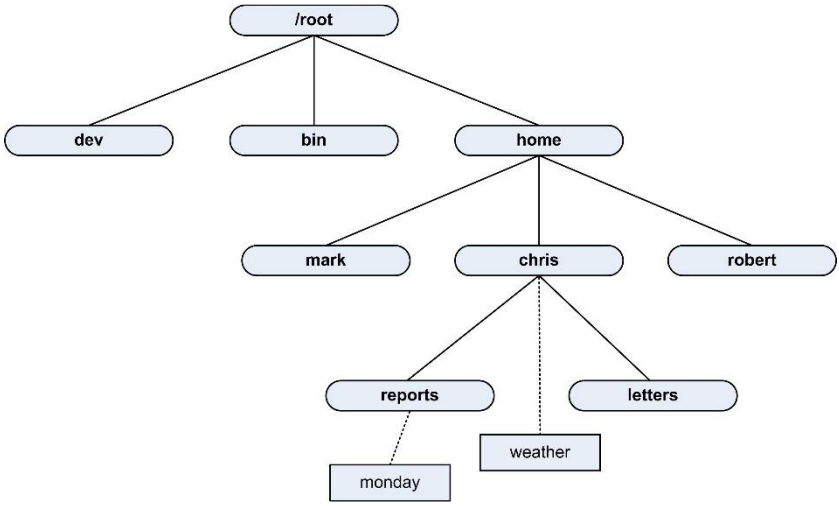

Рис.3. Путевое имя для файла weather: /home/chris/weather

Путевые имена могут быть полными и сокращенными. Полное путевое имя — это полное имя файла или каталога, начинающееся символом корневого каталога. Сокращенное путевое имя начинается символом рабочего каталога и представляет собой обозначение пути к файлу относительно вашего рабочего каталога. В структуре каталогов, изображенной на рис. 3, chris — рабочий каталог, а сокращенное путевое имя файла monday — reports/monday. Полное путевое имя этого файла — /home/chris/re-ports/monday.

Для упрощения доступа к своим файлам пользователь может указывать специальный символ «тильда» (~), который представляет полное путевое имя начального каталога.

Специальные символы

Существует три специальных символа заметно упрощающих использование консольных команд при работе с файловой системой. Их описание и несколько примеров использования приводятся ниже.

Специальные файловые символы:

- ~ - домашний каталог текущего пользователя. Неважно где вы находитесь в данный момент. Использование символа «~» позволит обратиться к любому файлу в личной директории.
- .. - каталог, располагающийся на уровень выше. Если вы находитесь в «/home/user-name/games», символ «..» будет указывать на «/home/user-name».
- . - текущий каталог. Использование аналогично символу «..», но не отбрасывает текущий уровень.

Расположение каталогов в файловой системе

Для того чтобы узнать, где располагается каталог, в котором мы сейчас находимся, используется команда:

```
$ pwd
```

Результат выполнения команды выглядит следующим образом:

```
/home/demo
```

Домашний каталог располагается после имени учетной записи пользователя, в приведенном примере он называется /demo. Этот каталог находится в каталоге с именем /home и в каталоге верхнего уровня, который называется "root" или корневой каталог, поэтому представлен одной косой чертой "/".

Просмотр содержимого каталогов

Для просмотра содержимого каталога используется команда **ls**. Название этой команды является сокращением слов list files.

Просмотреть содержимое любой директории можно следующими способами:

```
$ ls
```

$ ls имя_директории

В результате выполнения команды отображаются файлы и каталоги, находящиеся внутри данного каталога:

dir1 file1 file2 file3

К команде ls можно добавлять дополнительные флаги, например, чтобы показать детализированное представление (права, список владельцев файлов или папок, размер, дату последнего модифицирования) файлов и директорий в текущей директории, можно использовать флаг -l:

$ ls -l

Результат выполнения команды:

```
total 16
drwxr-xr-x 2 home demo 4096 Nov  3 17:40 dir1
-rw-r--r-- 1 home demo   13 Nov  8 17:17 file1
-rw-r--r-- 1 home demo   42 Nov  9 13:04 file2
-rw-r--r-- 1 home demo  42 Nov  10 13:04 file3
```

Для просмотра списка всех файлов, включая скрытые файлы и каталоги, вы можете добавить флаг -a:

$ ls -a

Результат выполнения команды:

```
.      dir1      file1          .mysql_history  .ssh
..     .bash_history  file2   file3  .nan
```

Отобразить содержимое текущей директории с добавлением к именам символов, характеризующих тип, можно с помощью команды:

```
$ ls -F
```

Результат выполнения команды:

```
dir1/   file1  file2  file3
```

Перемещение между каталогами

Для перехода в домашний каталог используется команда **cd**. Название этой команды является сокращением слов change directory.

Для перехода в домашний каталог пользователя user используется команда:

```
$ cd ~user
```

Для перехода в предыдущую директорию, в которой мы находились до перехода в текущую директорию также используется команда cd.

Для перехода в директорию уровнем выше используется команда:

```
$ cd ..
```

ОПЕРАЦИИ С ФАЙЛАМИ И КАТАЛОГАМИ

Создание и удаление каталогов

Пользователь может создавать и удалять собственные каталоги, а также изменять свой рабочий каталог. Каталоги создаются и удаляются, соответственно, командой **mkdir** и командой **rmdir**. В том и другом случае можно использовать путевые имена каталогов. В следующем примере пользователь сначала создает каталог reports, а затем, используя полное путевое имя, — каталог letters.

```
$ mkdir reports
$ mkdir /home/chris/letters
```

Для удаления каталога нужно применить команду rmdir с именем этого каталога. В приведенном ниже примере пользователь сначала удаляет командой rmdir каталог reports, а затем, указав полное путевое имя, — каталог letters.

```
$  rmdir reports
$  rmdir  /home/chris/letters
```

Команда rmdir позволяет удалить пустую папку. Если вам нужно удалить папку с файлами, то нужно использовать утилиту rm вместе с опцией -r.

Создание текстового файла и добавление в него информации.

Для создания простого текстового файла используется команда **cat** с символом перенаправления потока >. Для создания текстового файла введите: cat > имя_файла, далее необходимо ввести необходимый текст после этого нажать ctrl+Z или ctrl+C.

```
$ cat > file1
ordinary text
```

new line
last string

При использовании данной команды будьте внимательны, потому что последняя набранная строка не сохраняется в файл. Для ее сохранения просто переведите курсор на новую строку.

Для добавления строки в конец файла используется команда cat >> имя_файла.

$ cat >> file1
new string

Команда **touch** создает пустой файл. Также она позволяет указывать дату создания, права доступа и другие метаданные.

$ touch file

Опция -t позволяет установить дату создания. Дата указывается опцией -t в формате YYMMDDHHMM.SS. Если не указать, будет установлена текущая дата. Пример:

$ touch -t 201601081830.14 файл

Можно использовать дату создания другого файла:

$ touch -r шаблон файл

Также можно установить дату последней модификации, с помощью опции -m:

$ touch -m -t 201601081830.14 файл

Или дату последнего доступа:

$ touch -a -t 201601081830.14 файл

Чтобы посмотреть, действительно ли задаётся информация, которую вы указали, используйте команду stat:

$ stat файл

Отображение файлов

Во многих случаях бывает необходимо просматривать содержимое файла. Команды **cat** и **more** выводят содержимое файла на экран. Название команды cat образовано путем сокращения слова concatenate. В приведенном ниже примере осуществляется вывод текста файла на экран:

$ cat mydata
computers

Команда cat выводит на экран сразу весь текст файла. Если файл большой, то текст очень быстро мелькает на экране. Для устранения этого недостатка служит команда more, с помощью которой текст на экран можно выводить порциями. Эта команда вызывается с именем файла, который вы хотите просмотреть:

$ more mydata

Когда more вызывает файл, отображает его первый фрагмент, умещающийся на экране. Для отображения следующего фрагмента нажимается клавиша [f] или клавиша пробела. Для возврата к предыдущему тексту используется клавиша [Ь]. Нажав клавишу [q], можно в любой момент выйти из данной программы.

Команда **less** не входит в эталонную версию UNIX, но в семействе команд Slackware Linux она является чрезвычайно полезным пополнением. Команда less предоставляет дополнительные

возможности при просмотре файлов — а именно, возможность как прямого, так и обратного перемещения по файлу. Например, следующая команда просматривает файл с именем test:

```
$ less test
```

Большое преимущество less — обратное перемещение по файлу при нажатии клавиши b.

Очень большие файлы нежелательно загружать полностью или просматривать от начала до конца, особенно если вам лишь хочется бросить беглый взгляд на содержимое файла. В таких случаях можно воспользоваться командой **head** для просмотра начала файла или командой tail — для просмотра конца.

Обе команды по умолчанию отображают по 10 строк. Следовательно, следующая команда отображает первые 10 строк файла report:

```
$ head report
```

Для просмотра 10 последних строк того же файла используется другая команда:

```
$ tail report
```

Чтобы вывести другое количество строк, передайте его команде в качестве аргумента; например, следующая команда отображает первые 20 строк файла report:

```
$ head -20 report
```

Перемещение и копирование файлов

Чтобы создать копию файла, нужно указать команде **cp** в качестве параметров два имени файла. Первое из них — имя копируемого файла,

который уже существует. Этот файл часто называют исходным. Второе — имя, которое вы хотите присвоить копии. Это будет новый файл, содержащий копию всех данных исходного Файла. Его часто называют выходным файлом. Команда cp имеет следующий синтаксис: cp исходный_файл выходной_файл

В следующем примере пользователь копирует файл proposal в новый файл oldprop:

```
$ cp proposal oldprop
```

Может случиться так, что при копировании файла с помощью команды cp вы непреднамеренно уничтожите другой файл. При создании копии посредством этой команды сначала создается файл, а затем в него копируются данные. Если какой-нибудь файл уже имеет то же имя, которое вы указали для выходного файла, первый из них уничтожается и создаётся новый файл с этим именем. В большинстве дистрибутивов Linux существует возможность настроить систему для выявления подобных случаев. Если же такой возможности нет, используйте команду cp с опцией -i (сокращение от interactive). При использовании этой опции команда вначале проверяет, существует ли файл с указанным именем. Если да, то программа спросит, хотите ли вы перезаписать этот файл. Если вы ответите y, то существующий файл будет уничтожен и программа создаст новый файл в качестве копии. Если вы дадите другой ответ, он будет считаться отрицательным и выполнение команды cp будет прервано, а файл оригинала сохранен.

```
$ cp -i newprop proposal
Overwrite proposal? N
```

Для того чтобы скопировать файл из рабочего каталога в другой, нужно указать имя этого каталога команде cp в качестве второго параметра. Новая копия будет иметь то же имя, что и оригинал, но

разместится в другом каталоге. Файлы в разных каталогах могут иметь одинаковые имена. Поскольку файлы находятся в разных каталогах, они считаются разными. cp имена_файлов имя_каталога

Команда cp может использовать в качестве параметров имена многих файлов, заданные в виде списка, поэтому можно одновременно копировать в каталог сразу несколько файлов. Введите имена этих файлов в командной строке, причем имя каталога должно быть последним параметром. Все эти файлы копируются в указанный каталог. В следующем примере пользователь копирует файлы preface и docl в каталог props. Обратите внимание на то, что props — последний параметр.

```
$ cp preface docl props
```

При создании списка имен файлов для команды cp или команды mv можно использовать любые групповые символы. Пусть, например, вам нужно скопировать в заданный каталог все файлы с исходными текстами программ, написанными на языке C. Вместо того чтобы указывать в командной строке все эти файлы, можно ввести групповой символ * с расширением «.c», обозначая тем самым все файлы с расширением .c (то есть все файлы исходных текстов программ C), формируя таким образом их список. В следующем примере пользователь копирует все файлы исходных текстов программ из текущего каталога в каталог sourcebks.

```
$ cp *.c sourcebks
```

С помощью команды **mv** можно либо переименовать файл, либо переместить файл из одного каталога в другой. Используя mv для переименования файла, в качестве второго параметра нужно указать новое имя файла. Первый параметр — текущее имя файла. mv текущее_имя_файла новое_имя_файла

В следующем примере имя файла proposal меняется на versionl.

```
$ mv proposal version1
```

Как и при использовании команды ср, здесь очень легко ошибиться, удалив нужный файл. Переименовывая файл, вы можете выбрать имя, которое уже носит другой файл, и этот файл будет удален. Команда mv тоже имеет опцию -i, которая сначала проверяет, существует ли файл с указанным именем. Если да, программа спросит, хотите ли вы перезаписать его.

В следующем примере файл с именем version уже существует. Программа обнаруживает, что должна быть выполнена перезапись, и спрашивает, хотите вы это сделать или нет.

```
$ ls
Proposal versionl
$ mv -i versionl proposal
Overwrite proposal? n
```

Файл можно перенести из одного каталога в другой. Для этого нужно в качестве второго параметра в команде mv поставить имя каталога. В данном случае можно считать, что команда mv не переименовывает файл, а просто перемещает его из одного каталога в другой. После перемещения файла у него останется то имя, которое он носил в исходном каталоге (если вы не укажете иного). mv имя_файла имя_каталога

При создании списка имен файлов для команды mv можно использовать любые специальные символы. В приведенном ниже примере пользователь перемещает все файлы исходных текстов программ из текущего каталога в каталог newproj.

```
$ mv *.c newproj
```

Перемещение и копирование каталогов

Система Linux позволяет копировать и перемещать целые каталоги. В качестве первого параметра команды **cp** и **mv** могут использовать имя каталога, позволяя копировать и перемещать подкаталоги из одного каталога в другой. Первый параметр— имя перемещаемого или копируемого каталога, а второй— имя каталога, в который он будет помещен. При перемещении и копировании каталогов действует та же структура путевых имен, что и при соответствующих операциях с файлами.

Подкаталоги можно копировать из одного каталога в другой. Для копирования каталога команду cp необходимо использовать с опцией -r (сокращение от recursive, то есть «рекурсивный»). Эта опция дает команде cp указание копировать каталог вместе со всеми его подкаталогами. Другими словами, копируется все поддерево каталогов, начиная с указанного. В следующем примере каталог thankyou копируется в каталог oldletters. После завершения этой операции начинают равноправно сосуществовать два подкаталога thankyou: один в каталоге letters, другой в oldletters.

```
$ cp -r letters/thankyou oldletters
$ ls -F letters
/thankyou
$ ls -F oldletters
/thankyou
```

Удаление файла

Команда **rm** также позволяет удалять не только файлы, но и каталоги.

В следующем примере пользователь удаляет файл oldprop.

```
$ rm oldprop
```

Команда rm может быть использована с любым числом параметров, что позволяет одновременно удалять несколько файлов. Имена этих файлов указываются в командной строке после имени команды.

```
$ rm proposal version1 version
```

Командой rm следует пользоваться осторожно, так как отменить ее действие нельзя. Если файл удален, восстановить его не удастся. Чтобы избежать подобных ошибок, используйте команду rm с опцией -i, которая инициирует выдачу запроса на подтверждение удаления. Теперь перед удалением каждого файла система будет спрашивать, действительно та вы хотите удалить его. Если вы введете у, файл будет удален. При любом ином ответе файл не удаляется. В следующем примере посредством команды rm система получает указание удалить файлы proposal и oldprop, а затем запрашивает подтверждение по каждому из них. Пользователь решает удалить oldprop, а proposal оставить.

```
$ rm -i proposal oldprop
Remove proposal? n
Remove oldprop? У
```

Для удаления каталога и всех его подкаталогов применяется команда rm с опцией -r. Это очень мощная команда, и при ее неправильном использовании можно легко стереть все свои файлы. В следующем примере показано, как удалить каталог reports и все его подкаталоги:

```
$ rm -r reports
```

Следующая команда стирает все файлы в текущем каталоге.

```
$ rm *
```

Это очень мощная операция, позволяющая стирать целые сегменты файловых систем. Ею следует пользоваться с осторожностью.

Символические и жесткие ссылки linux

Символические и жесткие ссылки — это особенность файловой системы Linux, которая позволяет размещать один и тот же файл в нескольких директориях. Это очень похоже на ярлыки в Windows, так как файл на самом деле остается там же где и был, но вы можете на него сослаться из любого другого места.

В Linux существует два типа ссылок на файлы. Это символические и жесткие ссылки Linux. Они очень сильно отличаются и каждый тип имеет очень важное значение. В этой небольшой статье мы рассмотрим чем же отличаются эти ссылки, зачем они нужны, а также как создавать ссылки на файлы в Linux.

Символические ссылки более всего похожи на обычные ярлыки. Они содержат адрес нужного файла в вашей файловой системе. Когда вы пытаетесь открыть такую ссылку, то открывается целевой файл или папка. Главное ее отличие от жестких ссылок в том, что при удалении целевого файла ссылка останется, но она будет указывать в никуда, поскольку файла на самом деле больше нет.

Вот основные особенности символических ссылок:

- Могут ссылаться на файлы и каталоги;
- После удаления, перемещения или переименования файла становятся недействительными;
- Права доступа и номер inode отличаются от исходного файла;
- При изменении прав доступа для исходного файла, права на ссылку останутся неизменными;
- Можно ссылаться на другие разделы диска;
- Содержат только имя файла, а не его содержимое.

Жесткие ссылки. Этот тип ссылок реализован на более низком уровне файловой системы. Файл размещен только в определенном

месте жесткого диска. Но на это место могут ссылаться несколько ссылок из файловой системы. Каждая из ссылок — это отдельный файл, но ведут они к одному участку жесткого диска. Файл можно перемещать между каталогами, и все ссылки останутся рабочими, поскольку для них неважно имя.

Рассмотрим особенности:

- Работают только в пределах одной файловой системы;
- Нельзя ссылаться на каталоги;
- Имеют ту же информацию inode и набор разрешений что и у исходного файла;
- Разрешения на ссылку изменяться при изменении разрешений файла;
- Можно перемещать и переименовывать и даже удалять файл без вреда ссылке.

С помощью команды **ln** файлам можно присваивать дополнительные имена. Это нужно для того, чтобы иметь возможность обращаться к файлу по разным именам из разных каталогов. Дополнительные имена часто называют ссылками.

Команда ln принимает два параметра: имя исходного файла и новое, дополнительное имя файла. В операции ls указываются оба имени, но в действительности существует лишь один физический файл. ln исходное_имя_файла дололнительное_имя_файла

В следующем примере файлу today присваивается дополнительное имя weather. С его помощью также можно обратиться к файлу today.

```
$ ls
today
$ ln today weather
$ ls
today weather
```

Файлу можно дать несколько имен, применив ряд команд ln. В приведенном ниже примере файлу today присваиваются дополнительные имена weather и weekend.

```
$ ln today weather
$ ln today weekend
$ ls
today weather weekend
```

Используя команду ls с опцией -l, можно выяснить, есть ли у файла ссылки.

```
$ ls -l today weather
-rw-rw-r— 2 chris group 563 Feb 14 10:30 today
-rw-rw-r— 2 chris group 563 Feb 14 10:30 weather
```

Данные сведения, однако, не позволяют утверждать наверняка, что имена этих файлов связаны ссылками. Они просто показывают, что совпадает число ссылок, размеры и даты модификации двух файлов, как в случае с файлами today и weather. Для того чтобы знать это наверняка, нужно выдать команду ls с опцией -i. Эта команда сообщает имя файла и его индексный дескриптор. Индексный дескриптор — это уникальный номер которым система обозначает конкретный файл. Если индексные дескрипторы двух имен файлов совпадают, это значит, что они относятся к одному и тому же файлу. В следующем примере пользователь получает информацию о файлах today, weather и larisa. Обратите внимание на то, что today и weather имеют один и тот же индексный дескриптор.

```
$ ls -i today weather larisa
1234 today 1234 weather 3976 larisa
```

Чтобы удалить файл, нужно удалить все его ссылки. Имя файла фактически рассматривается как еще одна ссылка на этот файл, то

есть с помощью команды rm удаляется именно ссылка на тот или иной файл. Если ссылок было несколько и одна из них удаляется, к файлу можно обращаться по оставшимся даже в случае удаления исходной ссылки — первоначального имени файла. Это утверждение остается справедливым и после удаления первоначальной ссылки — исходного имени файла. В следующем примере файл today удаляется командой rm, но остается ссылка на этот файл с именем weather К файлу можно обращаться по этому имени.

```
$ ln today weather
$ rm today
$ cat weather
The storm broke today and the sun came out.
```

Ссылки, которые мы рассматривали до сих пор, называются прямыми ссылками (или жесткими ссылками — hard link). В принципе, в большинстве случаев удобно использовать именно прямые ссылки, но им присущ один серьезный недостаток: если вы попытаетесь создать ссылку на файл в каталоге другого пользователя, прямая ссылка может не сработать. Это обусловлено тем, что файловую структуру ОС Linux можно физически сегментировать на файловые системы. Файловая система может располагаться на любых физических запоминающих устройствах — от дискеты до комплекта жестких дисков. Несмотря на то, что файлы и каталоги во всех файловых системах присоединены к одному общему дереву каталогов, каждая файловая система физически управляет своими файлами и каталогами. Это значит, что файл одной файловой системы нельзя связать прямой ссылкой с файлом, принадлежащим другой файловой системе. Если вы попробуете создать ссылку на файл, находящийся в каталоге другого пользователя и принадлежащий другой файловой системе, прямая ссылка не сработает.

Символические ссылки создаются с помощью команды ln с опцией -s. В следующем примере пользователь создает ссылку lunch на файл /home/george/veglist.

```
$ ln -s /home/george/veglist lunch
```

Чтобы удалить файл, нужно удалить только прямые ссылки. Если остались символические ссылки, доступ к файлу по ним будет невозможен. В данном случае в символической ссылке содержится путевое имя уже не существующего файла.

В отличие от прямых ссылок, символические можно использовать для создания ссылок на каталоги. По сути дела, можно создать еще одно имя для обращения к каталогу. При этом следует помнить, что команда pwd всегда выдает фактическое имя каталога, а не символическое. В следующем примере пользователь создает для каталога thankyou символическую ссылку gifts. Используя ссылку gifts в команде cd, пользователь переходит в каталог thankyou. Команда pwd выдает путевое имя каталога thankyou.

```
$ ln -a /home/chris/letters/thankyou gifts
$ cd gifts
$ pwd
/home/chris/letters/thankyou
```

Рассмотрим опции утилиты:

-d — разрешить создавать жесткие ссылки для директорий суперпользователю;

-f — удалять существующие ссылки;

-i — спрашивать нужно ли удалять существующие ссылки;

-P — создать жесткую ссылку;

-r — создать символическую ссылку с относительным путем к файлу;

-s — создать символическую ссылку.

Поиск в каталогах

Команда **find** дает возможность искать файлы по имени, типу, владельцу и даже по времени последнего изменения. find список_каталогов -опция критерии Опция -name в качестве критериев задает образец и дает команде find указание искать имя файла, совпадающее с этим образцом. Для осуществления поиска файла по имени в команде find нужно задать имя каталога, опцию -name и имя файла. find список_каталогов -name имя_файла

В команде find используются также опции, которые просто вызывают выполнение какого-то действия, например вывод результатов поиска. Если вы хотите, чтобы команда find сообщила имена файлов, которые нашла, укажите в командной строке опцию -print. Эта опция дает указание вывести на стандартный вывод имена всех файлов, которые будут найдены с помощью команды find. В следующем примере пользователь ищет в каталоге reports все файлы с именем monday. После этого сокращенное путевое имя найденного файла выводится на экран.

```
$ find reports -name monday -print
reports/monday
```

Если вы собираетесь провести поиск в рабочем каталоге, замените его имя точкой. Двойная точка обозначает родительский каталог. В следующем примере производится поиск во всех файлах и подкаталогах рабочего каталога, причем имя рабочего каталога заменяется точкой. Если вы находитесь в своем начальном каталоге, то это — весьма удобная отправная точка для осуществления поиска во всех ваших каталогах. Найденные имена файлов выводятся с точкой.

```
$ find . -name weather -print
./weather
```

В качестве части образца при поиске файлов можно использовать специальные символы командного интерпретатора. Для того чтобы командный интерпретатор не обрабатывал специальные символы, не забудьте поставить кавычки. В следующем примере производится поиск всех файлов с расширением .c в каталоге programs.

```
$ find programs -name '*.c' -print
```

Команду find можно применять и для поиска каталогов. В ОС Linux каталог классифицируется как особый тип файла. Хотя все файлы имеют формат байтового потока, некоторые файлы, и каталоги в частности, используются особым образом. В этом смысле можно говорить о разных типах файлов. В команде find используется опция -type, с помощью которой обеспечивается поиск файла заданного типа. Опция -type имеет односимвольный управляющий параметр (модификатор), обозначающий тип файла. Каталог обозначается модификатором d. В приведенном ниже примере производится поиск каталога thankyou, для чего заданы его имя и тип d.

```
$ find /home/chris -name thankyou -type d -print
/home/chris/letters/thankyou
```

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Научиться использовать команды для работы с файлами в операционной системе Slackware Linux. Изучить основные ключи и параметры команд. Продемонстрировать работу команд:

1. Просмотра текущего рабочего каталога
2. Изменения рабочего каталога
3. Просмотра содержимого каталога (использовать различные ключи)
4. Создания каталогов
5. Удаления каталогов (пустого и не пустого)
6. Создания файлов (различными способами)
7. Редактирования текстовых файлов
8. Просмотра содержимого текстовых файлов (различными способами)
9. Отображения части текстовых файлов
10. Перемещения файлов и каталогов
11. Переименования файлов и каталогов
12. Копирования файлов и каталогов
13. Удаления файлов
14. Создания различных ссылок на файлы и каталоги
15. Поиска файлов в различных каталогах

При этом использовать специальные символы («.», «..», «~»), краткие и полные путевые имена. Для поиска и копирования файлов и каталогов продемонстрировать использование масок.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Опишите организацию файлов в ОС Linux.
2. Перечислите правила, используемые при именовании файлов в ОС Linux.
3. Проиллюстрируйте структуру файловой системы ОС Linux.
4. Раскройте понятие «начальный каталог».
5. Перечислите стандартные системные каталоги в ОС Linux.
6. Приведите классификацию путевых имен.
7. Опишите назначение специальных символов («.», «..», «~»).
8. Назовите команду для определения рабочего каталога.
9. Определите результат выполнения команды «$ ls -l».
10. Перечислите ключи команды ls.
11. Перечислите команды для создания и удаления каталогов.
12. Опишите особенности команды rmdir.
13. Предложите варианты использования команды mv.
14. Предложите вариант команды для удаления каталога и всех его подкаталогов.
15. Предложите вариант команды для поиска всех файлов с расширением «.txt» в рабочем каталоге.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания, ответы на контрольные вопросы, описание процесса выполнения лабораторной работы, выводы.

ОСНОВНАЯ ЛИТЕРАТУРА

1. Вирт, Н. Разработка операционной системы и компилятора. Проект Оберон [Электронный ресурс] / Н. Вирт, Ю. Гуткнехт ; пер.с англ. Борисов Е.В., Чернышов Л.Н.. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 560 с. — Режим доступа: https://e.lanbook.com/book/39992

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

2. Крищенко, В.А. Сервисы Windows [Электронный ресурс] : учебное пособие / В.А. Крищенко, Н.Ю. Рязанова. — Электрон. дан. — Москва : МГТУ им. Н.Э. Баумана, 2011. — 47 с. — Режим доступа: https://e.lanbook.com/book/52416..

3. Войтов, Н.М. Администрирование ОС Red Hat Enterprise Linux. Учебный курс [Электронный ресурс] : учебное пособие / Н.М. Войтов. — Электрон. дан. — Москва : ДМК Пресс, 2011. — 192 с. — Режим доступа: https://e.lanbook.com/book/1081

4. Стащук, П.В. Администрирование и безопасность рабочих станций под управлением Mandriva Linux: лабораторный практикум [Электронный ресурс] : учебно-методическое пособие / П.В. Стащук. — Электрон. дан. — Москва : ФЛИНТА, 2015. — 182 с. — Режим доступа: https://e.lanbook.com/book/70397

Электронные ресурсы:

5. Научная электронная библиотека http://eLIBRARY.RU
6. Электронно-библиотечная система http://e.lanbook.com
7. Losst - Linux Open Source Software Technologies https://losst.ru