

МДК.02.01 Программное обеспечение компьютерных сетей 3-курс

Занятие 07

Операции с дисковыми файлами

Операции с дисковыми файлами

Копирование файлов

Для копирования файлов используется команда `ср`.

Она позволяет копировать **файлы** или **каталоги**, допуская копирование **одного** файла в **другой**, а также копирование **группы файлов** в заданный **каталог**.

Синтаксис команды `ср` можно представить следующим образом:

```
ср [опции] файл путь
```

```
ср [опции] файл... каталог
```

Операции с дисковыми файлами

Копирование файлов

Если в качестве последнего параметра `ср` задан существующий **каталог**, то выполняется копирование исходных файлов в этот каталог с сохранением их имен.

В том случае, если параметрами являются **имена файлов**, `ср` копирует первый файл во второй.

Если командная строка содержит более двух параметров, не являющихся опциями самой команды, а последний параметр не является именем какого-либо каталога, то команда генерирует **ошибку**.

Операции с дисковыми файлами

Копирование файлов

Приведём следующий пример:

```
# cp text text2
```

Здесь файл `text` копируется в файл `text2`.

В том случае, если параметрами являются **имена файлов**, `cp` копирует первый файл во второй.

Попытка скопировать файл **сам в себя** ни к чему не приводит, кроме того, что выдается сообщение об **ошибке**.

Операции с дисковыми файлами

Копирование файлов

Если в качестве последнего параметра `ср` задан существующий **каталог**, то выполняется копирование исходных файлов в этот каталог с сохранением их имен.

Приведём следующий пример:

```
# cp text1 text2 DIR
```

Здесь файл `text1` и `text2` копируется в каталог `DIR`.

Опции `-R`, `-r` или `--recursive` используются при копировании директории.

В этом случае используется рекурсивное копирование — копируются директории и все их содержимое.

Операции с дисковыми файлами

Копирование файлов

Вот некоторые примеры использования команды `cp`.

Для копирования одного каталога в другой можно выполнить команду:

```
# cp -r DIR DIR.OLD
```

Здесь каталог `DIR` вместе со своим содержимым копируется в каталог `DIR.OLD`.

В следующем примере команда

```
# cp -r DIR1 DIR2 DIR12
```

копирует содержимое каталогов `DIR1` и `DIR2` в каталог `DIR12`.

Операции с дисковыми файлами

Перемещение файлов

Перемещение файлов в операционной системе UNIX выполняется с помощью команды `mv`, имеющей синтаксис:

```
mv [опции...] исходный_файл файл_назначения
```

```
mv [опции...] исходный_файл... каталог
```

Если последний параметр команды указывает на имя существующего **каталога**, то `mv` перемещает указанные **файлы** в этот каталог.

В том случае, если в качестве параметров заданы имена двух **файлов**, то имя первого файла будет **изменено** на имя второго.

Операции с дисковыми файлами

Перемещение файлов

Если же последний параметр не является **каталогом**, и заданы имена более чем **двух файлов**, то команда генерирует **ошибку**.

Когда *исходный_файл* и *файл_назначения* находятся в одной файловой системе, то изменяется **имя файла**, а **владелец, права доступа, атрибуты времени** остаются **неизменными**.

Если же они находятся в разных файловых системах, то *исходный_файл* **копируется** и затем **удаляется**.

Операции с дисковыми файлами

Перемещение файлов

Во время выполнения операции команда `mv` пытается скопировать время последней модификации, время доступа, идентификаторы пользователя и группы и права доступа к файлу.

Вот пример использования команды `mv`:

```
# mv test test.old
```

Здесь файл `test` переименоывается в файл `test.old`.

Ещё один пример использования команды `mv`:

```
# mv test DIR2
```

Здесь файл `test` перемещается в каталог `DIR2`.

Операции с дисковыми файлами

Удаление файлов и каталогов

Синтаксис этой команды таков:

```
rm -p [-s] каталог
```

Команда `rm` имеет две опции:

- `-p` — позволяет удалить **пустой** каталог вместе с его родительскими каталогами, отображая сообщение об успешном или неуспешном выполнении операции;
- `-s` — подавляет выдачу сообщений при использовании опции `-p`.

Операции с дисковыми файлами

Удаление файлов и каталогов

Команда `rm` функционирует иначе — с ее помощью можно удалить указанные **файлы**, но каталоги по умолчанию **не удаляются**.

При указании опций `-r` или `-R` будет удаляться все дерево каталогов нижезаданного каталога, включая и сам каталог, причем на глубину дерева не накладывается никаких ограничений.

Если последний компонент файла — символ `.` или `..`, то генерируется ошибка (это помогает избежать неприятных сюрпризов при выполнении команды `rm -r . *` или ей подобных).

Операции с дисковыми файлами

Удаление файлов и каталогов

Общие для UNIX-систем опции POSIX данной команды имеют следующий смысл:

- `-f` — не запрашивается подтверждение операции и не выдаются диагностические сообщения. При завершении команды с ошибками код ошибки не возвращается, если ошибки вызваны отсутствием файлов;
- `-i` — выводится запрос на подтверждение удаления (при указании опций `-f` и `-i` одновременно используется последняя);
- `-r` или `-R` — позволяет рекурсивно удалять дерево каталогов (с описанием).

Операции с дисковыми файлами

Удаление файлов и каталогов

Команда `rm` довольно **опасна!!!**

После ее выполнения **восстановить** удаленные файлы **невозможно**.

Поэтому нужно быть очень **внимательным** при ее использовании.

Вот пример использования команды `rmdir -p` :

```
# rmdir -p OLD
```

Здесь будет удалён каталог `OLD` в том случае, если он пустой.

Операции с дисковыми файлами

Удаление файлов и каталогов

Пример использования команды `rm` :

```
# rm test
```

Здесь будет удалён файл `test`.

Ещё один пример использования команды `rm` :

```
# rm -r DIR12
```

Здесь будет удалено всё дерево каталога `DIR12` на всю глубину со всем его содержимым.

Операции с дисковыми файлами

Создание каталогов

Создать новый каталог в UNIX можно с помощью команды `mkdir`.

В простейшем варианте команда использует один параметр (имя каталога), создавая каталог с указанным именем.

С помощью одной команды `mkdir` можно создавать несколько каталогов одновременно, перечисляя их в одной командной строке.

Синтаксис команды таков:

```
mkdir [опции] [список_каталогов]
```

Операции с дисковыми файлами

Создание каталогов

В команде `mkdir` можно использовать две опции:

- `-m` — позволяет задать в восьмеричной или символьной форме права доступа (как и для команды `chmod`), которые будут присвоены создаваемым каталогам;
- `-p` — кроме указанного каталога создаются любые требуемые промежуточные каталоги.

Если у пользователя нет прав на запись в родительский каталог, то новый каталог не создается, а если каталог уже существует (или файл с таким же именем), то команда генерирует ошибку.

Операции с дисковыми файлами

Создание каталогов

Пример использования команды `mkdir` :

```
# mkdir OLD
```

Здесь в текущей директории будет создан каталог `OLD`.

Ещё один пример использования команды `mkdir` :

```
# mkdir -m doc/test/2019
```

Здесь в текущей директории будет создан каталог со всеми входящими в него подкаталогами `doc/test/2019`.

Операции с дисковыми файлами

Поиск файлов и каталогов

Кроме копирования/перемещения и создания/удаления к часто выполняемым операциям с файлами и каталогами относится и ***поиск объектов файловой системы.***

Файловая система UNIX содержит десятки тысяч файлов, поэтому для быстрого поиска используются очень эффективные средства, одним из которых является команда `find`.

Команда имеет синтаксис:

```
find каталог ... выражение
```

Операции с дисковыми файлами

Поиск файлов и каталогов

Она просматривает иерархии каталогов в поисках файлов, удовлетворяющих критерию, задаваемому выражением выражение.

Выражения строятся из элементов с помощью следующих конструкций:

- `-name шаблон` — условие истинно, если имя файла соответствует шаблону.

При использовании метасимволов необходимо маскировать шаблоны от командного интерпретатора;

Операции с дисковыми файлами

Поиск файлов и каталогов

- `-type` *тип* — условие истинно, если файл — указанного типа.

Типы файлов задаются символами `b`, `c`, `d`, `f`, `l`, `p` и `s`, обозначающими, соответственно, специальное блочное устройство, специальное символьное устройство, каталог, обычный файл, символическую ссылку, именованный канал и сокет;

- `-user` *пользователь* — условие истинно, если файл принадлежит пользователю, указанному по идентификатору или регистрационному имени;

Операции с дисковыми файлами

Поиск файлов и каталогов

- `-group группа` — условие истинно, если файл принадлежит группе, указанной по идентификатору или имени;
- `-perm [-] права` — если дефис не задан, то условие истинно, только если права доступа в точности соответствуют указанным (как в команде `chmod`).

Если задан дефис, то условие истинно, если в правах доступа файла, как минимум, установлены те же биты, что и в указанных правах;

Операции с дисковыми файлами

Поиск файлов и каталогов

- `-size [+|-|=] n [c]` — условие истинно, если файл имеет длину n блоков (блок — 512 байтов) или символов (если указан суффикс c).

Перед размером можно указывать префикс `+` (не меньше), `-` (не больше) или `=` (в точности равен);

- `-atime [+|-|=] n` — условие истинно, если к файлу последний раз обращались n дней назад.

Перед n в элементах `-atime`, `-ctime` и `-mtime` можно указывать префикс `+` (не позже), `-` (не ранее) или `=` (ровно);

Операции с дисковыми файлами

Поиск файлов и каталогов

- `-ctime n` — условие истинно, если файл создан n дней назад;
- `-mtime n` — условие истинно, если файл был изменен n дней назад;
- `-newer файл` — условие истинно, если файл более новый, чем указанный;
- `-ls` — условие истинно всегда (выдает информацию о файле, аналогичную длинному листингу);

Операции с дисковыми файлами

Поиск файлов и каталогов

- `-depth` — условие истинно всегда — требует так обходить иерархию каталогов, чтобы файлы любого каталога всегда обрабатывались раньше, чем сам каталог (обход "в глубину");
- `-print` — условие истинно всегда (выдает полное имя файла в стандартный выходной поток).

В различных версиях операционной системы UNIX могут поддерживаться и другие компоненты выражений в команде `find`.

Если командная строка сформирована неправильно, команда немедленно завершает работу.

Операции с дисковыми файлами

Поиск файлов и каталогов

Вот несколько примеров использования команды `find`:

Пример 1.

Для отображения списка файлов текущего каталога программы достаточно выполнить команду:

```
# find . -print
```

Пример 2.

Для получения содержимого произвольного каталога, например, `/home/developer` нужно выполнить команду:

```
# find /home/developer -print
```

Операции с дисковыми файлами

Поиск файлов и каталогов

Пример 3.

Для поиска файлов в текущем каталоге с именами, которые заканчиваются на `tmp`, нужно выполнить команду:

```
# find . -name '*tmp' -print
```

Пример 4.

Здесь с помощью команды `find` выполняется поиск файлов с расширением `tmp` или `c`, находящихся в текущем каталоге:

```
# find . \( -name '*.tmp' -o -name '*.c' \) -print
```

Операции с дисковыми файлами

Поиск файлов и каталогов

В команде `find` можно задавать временные критерии поиска файлов, причем в самых различных комбинациях.

Следующий пример демонстрирует это: в нем используется опция `-atime [+|-|=] n`.

Условие является истинным, если время последнего доступа к файлу **больше/меньше**, чем $n*24$.

Например, команда `# find . \(-name '*.tmp' -o -name '*.pl' \) -atime +3 -print`

выполняет поиск файлов с указанными шаблонами, к которым не было обращения больше трех суток.

Операции с дисковыми файлами

Поиск файлов и каталогов

Нередко требуется найти файлы, принадлежащие определенному пользователю.

Например, следующая команда выполняет поиск файлов в каталоге `/usr`, владельцем которых является супер-пользователь `root`:

```
# find /usr -user root -print
```

Операции с дисковыми файлами

Поиск файлов и каталогов

Если критерием поиска является размер файла, то можно использовать следующую опцию: `-size` `[+|-|=]n[c]`.

Условие, задаваемое этой опцией, истинно, если размер файла **больше/меньше** *n*.

При этом различают два случая: если присутствует опция *c*, то размер файла предполагается заданным в байтах, если опция *c* отсутствует — то в блоках по 512 байтов.

Следующая команда выполняет поиск файлов, размер которых превышает 2048 байтов, в каталоге `/developer`:

```
$ find /developer -size +2048a -print
```

Операции с дисковыми файлами

Поиск файлов и каталогов

Расширить возможности команды `find` можно, перенаправив ее вывод не на стандартное устройство вывода, а в программный канал, как это показано в следующем примере:

```
# find TMP -name 't*' -print | grep tmp
```

Здесь команда `find` выполняет поиск файлов в каталоге `TMP`, удовлетворяющих шаблону `t*`, в имени которых присутствует `tmp`.

Операции с дисковыми файлами

Поиск файлов и каталогов

Конвейер программ чаще всего применяется в операциях копирования, перемещения и создания резервных копий файловых систем — при этом вывод команды `find` служит вводом для команды архивирования, как правило, `cpio`.

В следующем примере выполняется копирование файлов в другой каталог.

Операции с дисковыми файлами

Поиск файлов и каталогов

Конвейер программ чаще всего применяется в операциях копирования, перемещения и создания резервных копий файловых систем — при этом вывод команды `find` служит вводом для команды архивирования, как правило, `cpio`.

В следующем примере выполняется копирование файлов в другой каталог.

Операции с дисковыми файлами

Поиск файлов и каталогов

Для этого применяется команда `cpio -p`, которая принимает из стандартного входного потока список файлов и копирует или создает на них ссылки (опция `-l`) в каталоге `NEW`.

К моменту выполнения копирования каталоге `NEW` должен существовать.

Опция `-d` требует создания каталогов при необходимости.

Опция `-m` запрещает модификацию времени изменения файла.

Операции с дисковыми файлами

Поиск файлов и каталогов

Для генерации списка полных путей имен файлов для `cpio` в команде `find` нужно задать опцию `-depth`.

Это позволяет создавать файлы в каталогах, доступных **только для чтения**.

Вот так выглядит командная строка для выполнения операции копирования:

```
# find . -depth -print | cpio -pdlmv NEW
```

Операции с дисковыми файлами

Поиск файлов и каталогов

Заканчивая обзор возможностей операционной системы UNIX для работы с файлами, хочется добавить, что **дополнительную информацию** по данной теме можно почерпнуть:

- из справочных страниц, которые предоставляются почти всеми дистрибутивами, включая дистрибутивы ОС UNIX;
- или из многочисленных источников в **Интернете**.

Список литературы:

1. Unix и Linux: руководство системного администратора, 4-е издание, 2012, Э. Немерт, Г. Снайдер, Т. Хейн, Б. Уэйли
2. Для начинающих работать в UNIX, Ф.И. Торчинский.
3. Организация UNIX систем и ОС Solaris 9, Торчинский Ф.И., 2-е издание, исправленное, 2016.

Благодарю за внимание!

Преподаватель: Солодухин Андрей Геннадьевич

Электронная почта: asoloduhin@kait20.ru