

Лабораторная работа 4

АРХИВАЦИЯ И ПОИСК

Цель работы

Познакомиться с инструментами для работы с архивами. Получить представление о командах поиска, доступных пользователю командной строки.

1. Теоретические сведения

1.1. Поиск файлов

Несмотря на то, что существуют соглашения по организации файловой системы для *nix-подобных операционных систем, поиск необходимого файла может быть достаточно трудоемким.

В данной лабораторной работе мы рассмотрим несколько утилит предназначенных для поиска файлов в системе.

Команда **find** осуществляет рекурсивный обход дерева файловой системы, и производит поиск файла, основываясь на одном или нескольких атрибутах.

Таблица 1. Примеры опций команды **find**

Опция	Описание
-name	Имя файла
-type	Тип файла
-newer	Файлы с меньшей датой модификации чем у заданного файла
-mtime	Дата модификации файла
-size	Размер файла
-exec	Выполнение над найденными файлами указанной команды
-delete	Удаление найденных файлов

Некоторые опции могут принимать аргумент **n** со знаком. **+n** для значений больше **n**, **-n** для значений меньше **n**.

```
$ find ~ -name "*.txt" -type f -mtime -3
```

Приведенная выше команда произведет поиск всех обычных файлов в домашнем каталоге пользователя, с расширением **txt** и датой модификации меньше 3-х дней.

Команда **find** также поддерживает объединение шаблонов поиска в логическое выражение с помощью опций-операторов **-or**, **-and** и **-not**.

```
$ find ~ \( -size +100M \) -and \( -not -type d \)
```

Приведенная выше команда позволяет получить список файлов, не являющихся директорией размером больше 100 Мегабайт.

Альтернативным способом найти интересующий вас файл по имени является использование команды **locate**.

\$ locate имя_файла

Поиск в этом случае происходит не по дереву файловой системы, а по специальной базе имен файлов, которая периодически обновляется. Для обновления базы используется команда **updatedb**. По умолчанию **locate** не проверяет существуют ли файлы, найденные в базе имен на данный момент.

Команда **whereis** позволяет выполнить поиск расположения исполняемых файлов, файлов с исходным кодом и файлов справочных страниц для выбранной команды. Форматом вывода команды можно управлять с помощью специальных опций.

\$ whereis команда

Команда **which** выполняет поиск пути к исполняемому файлу для определенной команды на основе содержимого переменной **PATH**.

\$ which команда

Для поиска строк в файле содержащих определенный шаблон можно использовать команду **grep**. Если команде не передается список входных файлов, то поиск осуществляется в стандартном входном потоке.

```
$ cat /var/log/mylog | grep 'warning'
```

Таблица 2. Примеры опций команды **grep**.

Опция	Описание
-n	Вывод номера строки, содержащей шаблон
-c	Вывод количества строк, содержащих образец

-i	Игнорировать регистр символов
-v	Вывод всех строк, не содержащих шаблон

Утилиту **grep** часто используют в связке с другими командами, передавая ей данные на стандартный поток ввода. Возможно применение регулярных выражений при написании шаблона для поиска.

Команда **xargs** преобразует поступающие ей на вход строки в аргументы для заданной команды.

```
$ find ~/mydir -type f -name filename | xargs ls -l
```

1.2. Архивация

Для сокращения объема, занимаемого предназначенными для длительного хранения или передачи файлами, применяют различные утилиты сжатия.

Утилита **gzip** предназначена для сжатия одного или нескольких файлов. При упаковке оригинальные файлы заменяются файлами с расширением **.gz**. Перенаправить закодированную информацию на стандартный вывод с сохранением исходного файла можно с помощью опции **-c**.

```
$ gzip file1 file2 file3
```

Распаковка файлов может осуществляться с использованием опции **-d** или утилитой **gunzip**. Получить информацию о степени сжатия файла можно с помощью опции **-l**. С помощью числовых опций **-[1-9]** можно регулировать степень сжатия.

```
$ gzip -l file1.gz
```

Существует похожая на **gzip** утилита, использующая для сжатия преобразование Барроуза — Уилера, **bzip2**. Она имеет схожий синтаксис и опции. Сжатые файлы в этом случае имеют расширение **.bz2**.

Одновременно со сжатием, при работе с файлами применяется операция объединения нескольких файлов в один архив. *Архивация* часто применяется при создании резервных копий.

Утилита **tar** предназначена для упаковки множества файлов в архив, и их извлечения. В качестве аргументов команда принимает архивируемые файлы.

Таблица 3. Примеры опций команды **tar**.

Опция	Описание
-r	Добавление файлов к архиву
-c	Создание нового архива
--delete	Удаление файлов из архива
-t	Вывод содержимого архива
-x	Извлечение файлов из архива
-f	Использование архивного файла
-v	Вывод списка обрабатываемых файлов

Создание нового архивного файла:

```
$ tar -cvf myarchive.tar file1 file2 file3
```

Просмотр содержимого архивного файла:

```
$ tar -lf myarchive.tar
```

Утилита **zip** выполняет одновременно функции сжатия и архивации. Итоговый файл будет иметь расширение **.zip**. Для помещения в архив директорий вместе с вложенными файлами используется флаг **-r**.

```
$ zip -r archive.zip имя_директории
```

Для распаковки архива используется утилита **unzip**. Для получения дополнительной информации об извлекаемых файлах используется опция **-l**.

```
$ unzip -l archive.zip путь_для_распаковки
```

2 Порядок выполнения лабораторной работы

1. Прочитайте теоретический материал по лабораторной работе.
2. Ознакомьтесь с работой команд, приведенных в тексте лабораторной работы. Получите для них страницы справочного руководства.
3. С помощью утилит **find** и **wc** получите информацию о количестве файлов в домашнем каталоге пользователя.
4. Получите имена всех файлов, не являющихся символическими ссылками или каталогами, и поместите их в файл **filelist1.txt**.

5. С помощью команд **find**, **xargs** и **ls** получите полную информацию об атрибутах файлов домашнего каталога размер которых превышает 5 Килобайт и поместите результат в файл **filelist2.txt**.
6. С помощью команды **locate** получите список имен файлов содержащих в названии строку "**bash**".
7. Для команд, используемых в предыдущих подпунктах, получите расположения файлов справочных руководств.
8. Из файла **passwd_example** с помощью утилиты **grep**, получите записи пользователей с домашними каталогами в папке **home**, с указанием номеров строк. Поместите результат в файл **filelist3.txt**.
9. Сожмите файл **filelist1.txt** с сохранением исходного файла, утилитой **gzip** с различными степенями сжатия. Для получившихся файлов узнайте процент коэффициента сжатия.
10. Сожмите файл **filelist1.txt** с сохранением исходного файла, утилитой **bzip2** с различными степенями сжатия.
11. Сравните результаты для утилит **gzip** и **bzip2**.
12. Создайте архив **tar** содержащий файлы **filelist1.txt**, **filelist2.txt**, **filelist3.txt**.
13. Добавьте к созданному архиву файл **passwd_example**.
14. Создайте архив **zip** содержащий файлы **filelist1.txt**, **filelist2.txt**, **filelist3.txt** и **passwd_example**.
15. Сравните размеры получившихся архивов.
16. Распакуйте архив **blackbox.tar.gz** и запустите находящийся в нем скрипт.

3. Контрольные вопросы

1. Какие утилиты для поиска файлов вы знаете?
2. Как узнать расположения бинарных файлов определенной команды?
3. Где производит поиск файлов команда **locate**?
4. Как получить номера строк в файле не содержащие искомого шаблона?
5. Как добавить файлы к архиву **tar**, получить список файлов в архиве?
6. Как извлечь файлы из архива **tar**, **zip**?