Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

Направление 09.03.04 – «Программная инженерия»

Дисциплина: «Администрирование информационных систем»

Профиль: «Разработка программно-информационных систем»

Семестр 5

ОТЧЕТ

по лабораторной работе №4

Тема: «Архивация и поиск»

Выполнила: студент группы РИС-22-1б

Поважный В.Е. \_\_\_\_\_\_\_\_\_\_\_

Проверил: ст. преподаватель кафедры ИТАС

Шереметьев В. Г. \_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_

Пермь, 2024

**Цель работы**

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

**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 | Выополнение 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 логические 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

**Архивация**

Для сокращения объема, занимаемого предназначенными для долговременного 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 **-с**.

# $ gzip file1 file2 file3

Распаковка файлов может осуществляться c использованием опции **–d** или утилитой gunip. 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

**Порядок выполнения лабораторной работы**

1. С помощью утилит **find** и **wc** получите информацию о количестве файлов в домашнем каталоге пользователя.
2. Получите имена всех файлов, не являющихся символическими ссылками или каталогами, и поместите их в файл **filelist1.txt**.
3. 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**.
4. С помощью команды **locate** получите список имен файлов содержащих в названии строку “**bash”**.
5. Для команд, используемых в предыдущих подпунктах, получите расположения файлов справочных руководств.
6. Из файла **passwd\_example** c помощью утилиты **grep,** получите записи пользователей с домашними каталогами в папке **home,** c указанием номеров строк. Поместите результат в файл **filelist3.txt**.
7. Сожмите файл **filelist1.txt** с сохранением исходного файла, утилитой **gzip** c различными степенями сжатия. Для получившихся файлов узнайте процент коэффициента сжатия.
8. Сожмите файл **filelist1.txt** с сохранением исходного файла, утилитой

**bzip2** c различными степенями сжатия.

1. Сравните результаты для утилит **gzip** и **bzip2**.
2. Создайте архив **tar** содержащий файлы **filelist1.txt**. **filelist2.txt filelist3.txt**.
3. Добавьте к созданному архиву файл **passwd\_example**.
4. Создайте архив **zip** содержащий файлы **filelist1.txt**. **filelist2.txt**. **filelist3.txt** и **passwd\_example**.
5. Сравните размеры получившихся архивов.
6. Распакуйте архив **blackbox.tar.gz** и запустите находящийся в нем скрипт.

**Ход работы**

Чтобы получить информацию о количестве файлов в домашнем каталоге, можно использовать комбинацию утилит find и wc.

- **find ~ -type f**: эта команда ищет все файлы (-type f) в домашнем каталоге (~ указывает на домашний каталог текущего пользователя).

- **wc -l**: эта команда подсчитывает количество строк, которые поступили на вход, то есть количество файлов, найденных find.



Рис. 1 – количество всех найденных файлов

Для того чтобы получить имена всех файлов, которые не являются символическими ссылками или каталогами, и сохранить их в файл filelist1.txt

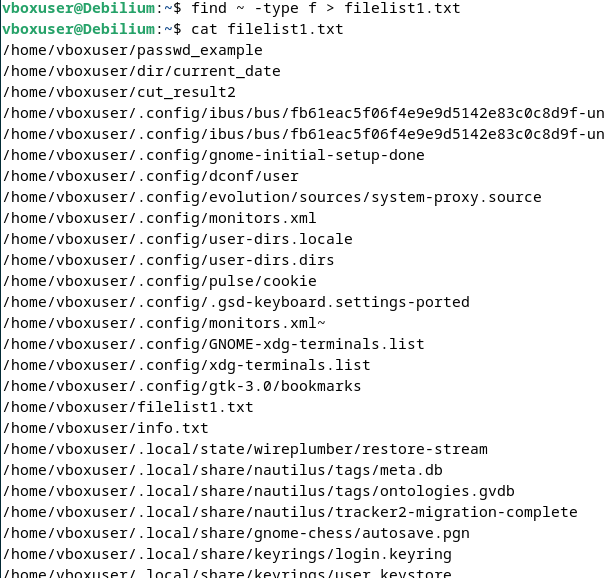


Рис. 2 – Запись всех файлов в файл.

Для получения полной информации об атрибутах файлов домашнего каталога, размер которых превышает 5 Кб.

- **-size +5k**: Указывает, что нужно искать файлы, размер которых превышает 5 Кб.

- **xargs ls -l**: Для каждого найденного файла xargs вызывает команду ls -l, которая выводит полную информацию об атрибутах файлов.

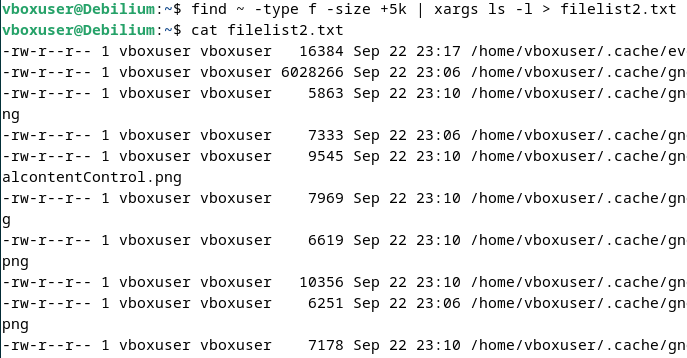


Рис. 3 – Запись всех файлов размером больше 5Кб в файл.



Рис. 4 – Вывод всех файлов с bash в названии.

Получение расположений файлов справочных руководств для указанных команд происходи при помощи опции –w для команды man.

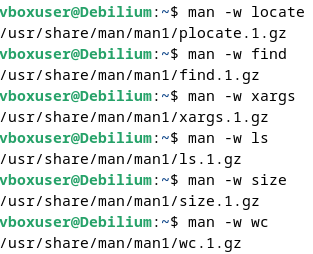


Рис. 5 – расположение файлов справочных руководств

Чтобы извлечь записи пользователей из файла passwd\_example, у которых домашние каталоги находятся в папке /home, и получить вывод с указанием номеров строк, можно выполнить следующую команду с использованием утилиты grep.

**grep -n '/home/' passwd\_example > filelist3.txt**

- **grep**: команда для поиска строк, соответствующих шаблону.

- **-n**: опция, указывающая на вывод номеров строк вместе с найденными строками.



Рис. 6 – записи пользователей.

Для сжатия файла filelist1.txt с помощью утилиты gzip с различными степенями сжатия и определения процентного коэффициента сжатия, нужно создать резервную копию при помощи команды cp

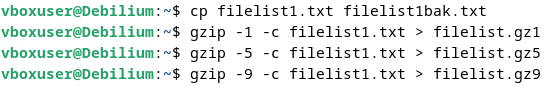


Рис. 7 – создание сжатых файлов.



Рис. 8 – коэффициент компрессии gzip.



Рис. 9 – Архивация файлов при помощи bzip2.



Рис. 10 – коэффициент компрессии bzip2.

В результате сжатия разными утилитами оказалось, что **bzip2** с разными степенями сжимает лучше **gzip**. **gzip** лучше подходит для ситуаций, где скорость важнее, а **bzip2** предпочтителен, когда важно максимально эффективное сжатие.

Чтобы создать архив tar, содержащий файлы filelist1.txt, filelist2.txt и filelist3.txt, вам нужно использовать команду tar.

- **-c** — опция для создания нового архива.

- **-v** — опция для отображения процесса (verbose), т.е. показывает, какие файлы добавляются в архив.

- **-f** — опция для указания имени создаваемого архива (в данном случае archive.tar).

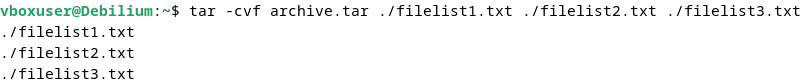


Рис. 11 – архивация 3-х файлов в tar архив

Чтобы добавить файл passwd\_example к уже существующему архиву archive.tar, нужно использовать опцию –r.

- **-r** — опция для добавления файлов в существующий архив.

****

Рис. 12 – добавление файла в архив.

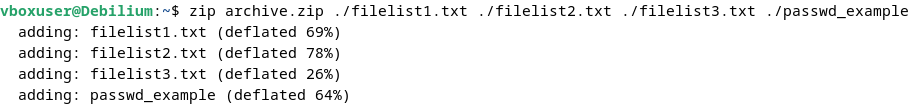
****

Рис. 13 – создание zip архива.

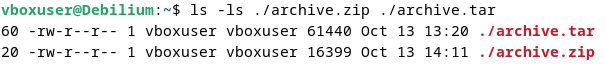
****

Рис. 14 – сравнение tar и zip архивов.

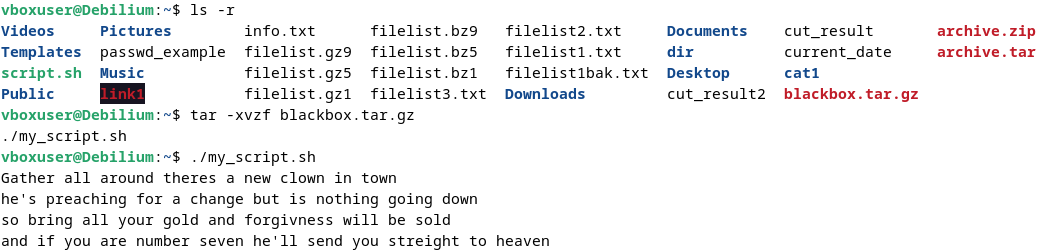


Рис. 14 – запуск скрипта из архива.

**Ответы на контрольные вопросы**

1. Утилиты для поиска файлов:

- `find` — поиск файлов по имени, дате изменения, размеру, правам доступа и другим атрибутам.

- `locate` — быстрый поиск по индексированной базе данных файлов.

- `whereis` — поиск бинарных файлов, исходных текстов и страниц справки для команд.

- `which` — поиск расположения исполняемого файла команды.

- `grep` — поиск текстовых шаблонов в файлах.

- `fd` — быстрая альтернатива `find` с удобным синтаксисом.

2. Как узнать расположение бинарных файлов команды?

- Используйте команду `which` для определения пути исполняемого файла:

which command\_name

- Можно также использовать `whereis`:

whereis command\_name

- Если нужно больше деталей, включая все пути, где бинарный файл может находиться:

type -a command\_name

3. Где производит поиск файлов команда `locate`?

- Команда `locate` ищет файлы в базе данных `/var/lib/mlocate/mlocate.db`. Эта база обновляется утилитой `updatedb`, которая запускается периодически или вручную для индексации файловой системы.

4. Как получить номера строк в файле, не содержащие искомого шаблона?

Для этого используется команда `grep` с флагом `-v` (вывести строки, не содержащие шаблона) и `-n` (отображение номеров строк):

grep -n -v "шаблон" filename

5. Как добавить файлы к архиву TAR и получить список файлов в архиве?

- Добавить файлы к существующему архиву TAR:

tar -rvf archive.tar newfile1 newfile2

Флаг `-r` используется для добавления файлов в архив.

- Получить список файлов в архиве TAR:

tar -tvf archive.tar

6. Как извлечь файлы из архива TAR и ZIP?

- Извлечение файлов из архива TAR:

tar -xvf archive.tar

Если это архив сжатый с помощью GZIP (например, `.tar.gz`):

tar -xvzf archive.tar.gz

- Извлечение файлов из архива ZIP:

unzip archive.zip

Эти команды извлекут все файлы из соответствующих архивов в текущую директорию.