Лабораторная работа №4. Перенаправление потоков и командыфильтры

Цель: изучение способов перенаправления ввода-вывода команд, организации конвейеров команд и использования в конвейерах команд-фильтров.

ЗАДАНИЕ

1. Изучите главу 5 теоретического курса «Перенаправления ввода-вывода и командыфильтры».
2. Войдите в Linux и запустите программу Terminal.
3. Выполните на рабочем компьютере все практические примеры из данной главы.
4. Создайте текстовый файл содержащий таблицу данных о пользователях. Информация о каждом пользователе хранится в одной строке и включает в себя следующие данные (поля): ФИО, телефон с кодом города, домашний адрес, адрес электронной почты. Разделителем полей является символ «точка с запятой». Выведите информацию обо всех пользователях с именем Сергей. Выведите количество пользователей с телефонами, для которых указан код города 495. Выведите информацию обо всех пользователях, имеющих электронную почту на сервере Google. Выведите информацию о первом пользователе, проживающем в Туле.
5. Создайте текстовый файл, содержащий строки вида «имя=значение» (имена могут повторяться). Составьте команду, определяющую по заданному имени, соответствующее ей значение из правой части строки. Модифицируйте команду для поиска N-го по порядку значения в файле для заданного имени. Составьте команду, подсчитывающую количество строк в файле с одним и тем же именем слева. Выведите все строки файла для заданного имени, упорядоченные по значению (рассмотрите варианты как текстовых, так и числовых значений).
6.Напишите команду, выводящую все строки файла, содержащие IP-адреса. Проверьте команду на следующих примерах правильных и неправильных IP-адресов (каждое из четырех полей, разделенных точками, должно содержать число от 0 до 255).
Правильные IP-адреса: 127.0.0.1 192.168.1.1 192.168.25.249 192.168.0.255
Неправильные IP-адреса: 12345.0.0.1 192.168. 258.1 345.168.1.1

```
guzovskiy@debian:~$ cat > user.txt <<E0F
> Сергей Попов;4953243546;Москва, ул. Попова, д.4;popov@gmail.com
> Елизавета Менькова;54927685491;Санкт-Петербург, ул. Меньковой, д.5;lizok228@ma
il.com
> Сергей Бабков;4959944551; Тула, ул. Бабкова, д.6;anigilator2009@gmail.com
> Олег Ничипоренко;4952282223;Тула, ул.Ничипоренко,д.7;kizaruoleg@gmail.com
> E0F
```

Рисунок 1 - Создание файла users.txt

```
guzovskiy@debian:~$ grep "^Сергей" user.txt
Сергей Попов;4953243546;Москва, ул. Попова, д.4;popov@gmail.com
Сергей Бабков;4959944551; Тула, ул. Бабкова, д.6;anigilator2009@gmail.com
```

Рисунок 2 - Вывод пользователей с именем Сергей

```
guzovskiy@debian:~$ grep ";495" user.txt | wc -l
3
```

Рисунок 3 - Подсчёт пользователей с кодом города 495

```
guzovskiy@debian:~$ grep "@gmail.com" user.txt
Сергей Попов;4953243546;Москва, ул. Попова, д.4;popov@gmail.com
Сергей Бабков;4959944551; Тула, ул. Бабкова, д.6;anigilator2009@gmail.com
Олег Ничипоренко;4952282223;Тула, ул.Ничипоренко,д.7;kizaruoleg@gmail.com
```

Рисунок 4 - Пользователи с почтой на gmail

```
guzovskiy@debian:~$ grep "Тула" user.txt | head -n 1
Сергей Бабков;4959944551; Тула, ул. Бабкова, д.6;anigilator2009@gmail.com
```

Рисунок 5 - Первый пользователь из Тулы

```
guzovskiy@debian:~$ cat > values.txt <<E0F
> key1=100
> key2=200
> key1=300
> key3=400
> key1=150
> E0F
```

Рисунок 6 - Создание файла values.txt

```
guzovskiy@debian:~$ grep "^key1=" values.txt | cut -d'=' -f2
100
300
150
```

Рисунок 7 - Получить значение по имени

```
guzovskiy@debian:~$ grep "^key1=" values.txt | cut -d'=' -f2 | sed -n '2p'
300
```

Рисунок 8 - Получить N-е значение

```
guzovskiy@debian:~$ cut -d"=" -f1 values.txt | sort | uniq -c
      3 key1
      1 key2
      1 key3
```

Рисунок 9 - Подсчитать количество строк с одинаковым именем

```
guzovskiy@debian:~$ grep "^key1=" values.txt | sort -t'=' -k2
key1=100
key1=150
key1=300
```

Рисунок 10 - Упорядочить строки по значению (текстовое сравнение)

```
guzovskiy@debian:~$ grep "^key1=" values.txt | sort -t'=' -k2n
key1=100
key1=150
key1=300
```

Рисунок 11 - Упорядочить строки по числовому значению

```
guzovskiy@debian:~$ cat > ips.txt <<E0F
> 127.0.0.1
> 192.168.1.1
> 192.168.25.249
> 192.168.0.255
> 12345.0.0.1
> 192.168.
> 258.1
> 345.168.1.1
> E0F
```

Рисунок 12 - Пример файла ips.txt

```
guzovskiy@debian:~$ grep -E '^([0-9]{1,3}\.){3}[0-9]{1,3}$' ips.txt | awk -F. '$1<=255
&& $2<=255 && $3<=255 && $4<=255'
127.0.0.1
192.168.1.1
192.168.25.249
192.168.0.255
```

Рисунок 13 - Команда для поиска валидных IP-адресов

Контрольные вопросы

1. В операционной системе Linux существует три стандартных потока: стандартный ввод (stdin), стандартный вывод (stdout) и стандартный вывод ошибок (stderr). Эти потоки доступны каждому процессу с момента его запуска. По умолчанию ввод осуществляется с клавиатуры, а вывод и ошибки отображаются на экране. Однако с помощью механизма перенаправления можно изменить направление этих потоков: например, перенаправить вывод команды в файл, а ошибки — в отдельный лог. Для перенаправления стандартного вывода используется символ >,

который записывает результат в указанный файл, перезаписывая его. Если нужно добавить данные в конец файла, применяется >>. Стандартный ввод можно перенаправить с помощью <, а ошибки — через 2>, где 2 обозначает файловый дескриптор stderr. Чтобы объединить вывод и ошибки в один файл, используется 2>&1 или современный вариант &>.

2. Команда wc предназначена для подсчёта строк, слов, байт и длины самой длинной строки в тексте. Она полезна для быстрой оценки объёма данных. Команда tee позволяет одновременно выводить данные на экран и записывать их в файл, что удобно при построении конвейеров.

3. Команды head и tail используются для вывода начала и конца файла соответственно. С их помощью можно просматривать первые или последние строки, а также следить за обновлением логов в реальном времени с помощью параметра -f.

4. Команда sort сортирует строки текста. Она поддерживает сортировку по алфавиту, по числовым значениям, по конкретным полям и в обратном порядке. Можно указать разделитель полей и удалить дубликаты.

5. Команда cut извлекает части строк — например, конкретные поля, разделённые символом ;, или диапазоны символов по позиции.

6. Для сравнения файлов используются команды cmp и diff. cmp сравнивает файлы побайтно и сообщает первое различие, а diff показывает все отличия построчно, что удобно для анализа изменений. Команда patch применяется для внесения изменений, созданных с помощью diff.

7. Команда egrep — это расширенная версия grep, поддерживающая расширенные регулярные выражения. Она позволяет искать строки по шаблонам, игнорировать регистр, искать целые слова или исключать совпадения.

8. Регулярные выражения используют специальные символы, называемые метасимволами. Символ . обозначает любой символ, ^ — начало строки, $ — конец строки, * — повторение от нуля и более, + — от одного и более, ? — ноль или одно повторение. Квадратные скобки [] задают набор символов, а [^] — исключение. Символ | используется для логического ИЛИ, а скобки () — для группировки. Конструкция {n,m} задаёт количество повторений от n до m.