

ЛАБОРАТОРНАЯ РАБОТА № 1.2.

Командный язык и скрипты LINUX ч.2.

Цель. Закрепить на практике основные команды для работы с фильтрами и системными утилитами.

Краткие теоретические сведения.

Рассмотрим команду `cut`. Чтобы пояснить действие этой команды запишем в файл `basa.txt` следующие строки:

```
Petrov: 20: 150: male  
Sidorov: 22: 50: male  
Nutrina: 21: 70: female  
Atomova: 19: 50: female
```

Мы видим, что записи разделены двоеточиями. Такие символы называют символами-разделителями (delimiters). Наш файл можно рассматривать как базу данных. Команда `cut` позволяет вывести столбцы, которые нас интересуют. Пример.

```
ovgerman@ovgerman:~$ cut -d":" -f1,3 basa.txt  
petrov:150  
sidorov:50  
nutrina:70  
utova:110  
ovgerman@ovgerman:~$
```

Здесь в команде `cut` указано, что делимитером (d) является знак двоеточия и что мы хотим вывести столбцы первый и третий (через запятую).

Следующий пример использует встроенный редактор строк (`sed`), который позволяет выполнить некоторые преобразования «на лету».

```
ovgerman@ovgerman:~$ cut -d":" -f1,3 basa.txt | sed 's:/ /g'  
petrov 150  
sidorov 50  
nutrina 70  
utova 110  
ovgerman@ovgerman:~$
```

К предыдущей команде мы здесь добавили

`| sed 's:/ /g'`

Первая буква **s** указывает, что будет производиться замена в строках. Далее указываем, что на что изменяет, а именно: заменяем на пустой символ. Буква **g** означает, что замена производится везде по всему тексту (global). Команда `sed '/50/d' file2` удаляет строки, где содержится цифра 50.

```
Petrov:20:150:male
Sidorov:22:50:male
Sedina :21:70:female
ovgerman@ovgerman:~$ sed '/50/d' file2
Nutrina:21:70:female
ovgerman@ovgerman:~$
```

Команда **grep** ищет фрагмент текста, заданного образцом или выражением. Например, пусть имеется файл, содержащий следующие строки:

```
...$ cat tennis.txt
Amelie Mauresmo, Fra
Kim Clijsters, BEL
Justine Henin, Bel
Serena Williams, usa
Venus Williams, USA
...$ cat tennis.txt | grep Williams
Serena Williams, usa
Venus Williams, USA
```

Здесь команда `grep` позволяет вывести людей по фамилии Williams.

Или

```
..$ grep Williams tennis
Serena Williams, usa
Venus Williams, USA
```

Можно использовать **grep** без команды **cat**.

Наконец, `grep $x file1.txt` – выводит значение переменной **x** из файла.

Задание 1. Используя последний пример, заменить **nutrina** на **seldina**.

Задание 2. Написать скрипт для вывода строки из файла **basa.txt**, запросив фамилию с консоли.

Задание 3. Запросить фамилию, вывести фамилию и зарплату.

Задание 4. Ввести фамилию, вывести только зарплату.

Создадим два текстовых файла (см. скриншот А ниже): **t1.txt** , **t2.txt**. У них есть общие элементы, есть и различные. Команда `comm` позволяет показать, какие элементы есть у первого файла, которых нет у второго; какие элементы

есть у второго файла, которых нет у первого, и какие элементы общие. Но для этого надо сначала отсортировать файлы. Пример дан на скриншоте В.

```
File Edit View Terminal Help
ovgerman@ovgerman:~$ touch t1.txt
ovgerman@ovgerman:~$ touch t2.txt
ovgerman@ovgerman:~$ echo one>>t1.txt
ovgerman@ovgerman:~$ echo two>>t1.txt
ovgerman@ovgerman:~$ echo four>>t1.txt
ovgerman@ovgerman:~$ echo seven>>t1.txt
ovgerman@ovgerman:~$ echo two>>t2.txt
ovgerman@ovgerman:~$ echo four>>t2.txt
ovgerman@ovgerman:~$ echo six>>t2.txt
ovgerman@ovgerman:~$ cat t1.txt
one
two
four
seven
ovgerman@ovgerman:~$ cat t2.txt
two
four
six
ovgerman@ovgerman:~$
```

Скриншот А

```
ovgerman@ovgerman:~$ sort t1.txt>t11.txt
ovgerman@ovgerman:~$ cat t11.txt
four
one
seven
two
ovgerman@ovgerman:~$ touch t22.txt
ovgerman@ovgerman:~$ sort t2.txt>>t22.txt
ovgerman@ovgerman:~$ cat t22.txt
four
six
two
ovgerman@ovgerman:~$
```

Скриншот В

Теперь можно выполнить команду comm:

```
ovgerman@ovgerman:~$ comm t11.txt t22.txt
      four
one
seven
      six
      two
ovgerman@ovgerman:~$
```

Можно вывести не все три столбца, а только один или два. Для этого в команде нужно указать, какие столбцы исключить. Так, выведем столбец общих элементов

```
ovgerman@ovgerman:~$ comm -12 t11.txt t22.txt  
four  
two  
ovgerman@ovgerman:~$
```

Здесь указано, что столбцы 1 и 2 выводить не надо. Будет выведен только третий столбец общих элементов.

Задание 5. Отсортировать файл. Т.е. файл должен сохранить свое собственное имя, но содержать отсортированные данные.

(Решение. Нужно предварительно создать промежуточный отсортированный файл t11.txt. Затем выполнить такие команды

Задание 6. Подсчитать число общих слов в обоих файлах. Для подсчета числа слов в файле использовать команду

\$wc -w file1

Задание 7. Написать скрипт, который создает отсортированный файл, содержащий слова из двух файлов, исключая их общую часть.