

Министерство образования Республики Беларусь
ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Кафедра технологий программирования

**Методические указания для выполнения
лабораторной работы № 3
по курсу «Операционные системы и системное
программирование»**

**«Файловая система: редактирование текстовых
файлов»**

Полоцк, 2019

Цель работы:

Освоить навыки работы с текстовыми файлами в ОС Linux;

План работы:

1. Создать текстовый файл вида <email> <имя> <фамилия> при помощи команды cat;
2. Создать текстовый файл вида <email> <номер телефона> при помощи редактора vi;
3. Объединить эти файлы по общему ключу <email> и вывести на экран эту выборку;
4. Произвести выборку из объединенного вывода файлов по столбцам <имя> <фамилия> <номер телефона>;
5. Дописать в каждый файл по две дублирующие (то или иное поле) записи посредством команды echo;
6. Отобразить только уникальные записи произвольного файла при помощи команды uniq;

Ход работы:

1. Команда cat — простой способ создавать текстовые файлы и выводить их на экран. Изначально нужна для конкатенации (объединения нескольких файлов).

Создадим произвольный файл при помощи cat:

```
# cat << EOF > file1.txt
> aaa@mail.ru Dmitry Petrov
> bbb@mail.ru Eugeny Ivanov
> ccc@mail.ru Pavel Smirnov
> EOF
```

Здесь мы обозначили, что текст будет вводится до появления метки EOF в стандартном вводе, а записываться всё будет в файл file1.txt.

Теперь вводим файл построчно. После ввода строки EOF (метка окончания файла) он будет записан на диск.

Командой ls мы убедимся, что в нашем каталоге присутствует файл *file1.txt*.

А затем командой

```
# cat file1.txt
```

выведем его содержимое на экран.

```
[root@localhost ~]# cat << EOF > file1.txt
> aaa@mail.ru Dmitry Petrov
> bbb@mail.ru Eugeny Ivanov
> ccc@mail.ru Pavel Smirnov
> EOF
[root@localhost ~]# ls
1.txt          Desktop        install.log    scsrn.log
anaconda-ks.cfg file1.txt      install.log.syslog TEST
[root@localhost ~]# cat file1.txt
aaa@mail.ru Dmitry Petrov
bbb@mail.ru Eugeny Ivanov
ccc@mail.ru Pavel Smirnov
[root@localhost ~]# _
```

2. Теперь создадим файл при помощи текстового редактора vi. Это очень мощный редактор и мы познакомимся с его основными возможностями несколько позже. Ну а пока:

```
# vi file2.txt
```

И переходим в режим вставки, нажав клавишу i.

Теперь можно набрать текст:

```
aaa@mail.ru +79612323232
```

```
bbb@mail.ru +79623131313
```

```
ccc@mail.ru +79634578787
```

Теперь нажимаем клавишу <ESC> чтобы выйти из режима вставки текста и вводим команду :wq чтобы записать файл на диск (w) и выйти (q)

```
aaa@mail.ru +79612323232
bbb@mail.ru +79623131313
ccc@mail.ru +79634578787
```

3. Объединяем эти файлы по общему ключу. Для объединения файлов служит команда join, которая по умолчанию принимает первые столбцы за ключевые (наш случай), поэтому просто выполняем команду:

```
# join file{1,2}.txt
```

или (одинаково)

```
# join file1.txt file2.txt
```

```
[root@localhost ~]# join file{1,2}.txt
aaa@mail.ru Dmitry Petrov +79612323232
bbb@mail.ru Eugeny Ivanov +79623131313
ccc@mail.ru Pavel Smirnov +79634578787
[root@localhost ~]# _
```

Как видим, файлы объединились в один, по общему ключу с электронной почтой. Если нужно указать другой столбец для объединения, то нужны ключи -1 <N> для первого файла и -2 <M> для второго. Весьма удобно!

4. Теперь представим, что у нас есть полный вывод, но нужны не все столбцы! В данном случае нужны только имя, фамилия и номер телефона. На помощь придет команда cut.

Скорим вывод предыдущей команды на ввод cut и укажем разделитель (-d «) — пробел и перечислим поля (-f) которые нам нужны, в данном случае 2, 3 и 4-ое поля.

```
# join file{1,2}.txt | cut -d " " -f2,3,4
```

```
[root@localhost ~]# join file{1,2}.txt | cut -d " " -f 2,3,4
Dmitry Petrov +79612323232
Eugeny Ivanov +79623131313
Pavel Smirnov +79634578787
[root@localhost ~]# _
```

Всё просто.

5. При помощи echo дозаписать в каждый файл по паре записей. Здесь ОЧЕНЬ важно обратить внимание на один нюанс. Команда echo выводит строку на устройство вывода. По умолчанию это экран. Можно ей указать не экран, а файл, и тогда строка запишется в этот файл. Если просто указать перенаправление >, то эта

строка заменит содержимое файла. А если двойное — >> , то строка дозапишется в конец. Нам нужно именно двойное.

```
# echo bbb@mail.ru Sergey Kalinin >> file1.txt
# echo ddd@mail.ru Dmitry Sverchkov >> file1.txt
# echo ccc@mail.ru +79658745963 >> file2.txt
# echo jjj@mail.ru +79632147852 >> file2.txt
```

Выведем результирующие файлы командой cat

```
[root@localhost ~]# cat file{1,2}.txt
aaa@mail.ru Dmitry Petrov
bbb@mail.ru Eugeny Ivanov
ccc@mail.ru Pavel Smirnov
bbb@mail.ru Sergey Kalinin
ddd@mail.ru Dmitry Sverchkov
aaa@mail.ru +79612323232
bbb@mail.ru +79623131313
ccc@mail.ru +79634578787
ccc@mail.ru +79658745963
jjj@mail.ru +79632147852
[root@localhost ~]# _
```

6. Итак, у нас есть дублированные поля. В Linux есть возможность найти как повторяющиеся поля, так и только уникальные (правда выборка при этом должна быть отсортирована). Итак, реализуем это:

```
[root@localhost ~]# cat file1.txt
aaa@mail.ru Dmitry Petrov
bbb@mail.ru Eugeny Ivanov
ccc@mail.ru Pavel Smirnov
bbb@mail.ru Sergey Kalinin
ddd@mail.ru Dmitry Sverchkov
[root@localhost ~]# cat file1.txt | sort
aaa@mail.ru Dmitry Petrov
bbb@mail.ru Eugeny Ivanov
bbb@mail.ru Sergey Kalinin
ccc@mail.ru Pavel Smirnov
ddd@mail.ru Dmitry Sverchkov
[root@localhost ~]# cat file1.txt | sort -k 2
aaa@mail.ru Dmitry Petrov
ddd@mail.ru Dmitry Sverchkov
bbb@mail.ru Eugeny Ivanov
ccc@mail.ru Pavel Smirnov
bbb@mail.ru Sergey Kalinin
[root@localhost ~]# _
```

Выведем сперва содержимое файла file1.txt:

```
# cat file1.txt
```

Затем отсортируем этот вывод:

```
# cat file1.txt | sort
```

Мы видим, что строки отсортированы по первому столбцу. Если нужно указать другой столбец, используем ключ -k <N>.

```
# cat file1.txt | sort -k 2
```

Этот пример сортирует по имени (второй столбец). Команда sort имеет много различных ключей. Всё есть в справке.

Ну чтоже, а теперь выведем дубликаты и только уникальные поля:

```

[root@localhost ~]# cat file1.txt | sort
aaa@mail.ru Dmitry Petrov
bbb@mail.ru Eugeny Ivanov
bbb@mail.ru Sergey Kalinin
ccc@mail.ru Pavel Smirnov
ddd@mail.ru Dmitry Sverchkov
[root@localhost ~]# cat file1.txt | cut -d" " -f1 | sort
aaa@mail.ru
bbb@mail.ru
bbb@mail.ru
ccc@mail.ru
ddd@mail.ru
[root@localhost ~]# cat file1.txt | cut -d" " -f1 | sort | uniq
aaa@mail.ru
bbb@mail.ru
ccc@mail.ru
ddd@mail.ru
[root@localhost ~]# cat file1.txt | cut -d" " -f1 | sort | uniq -d
bbb@mail.ru
[root@localhost ~]# cat file1.txt | cut -d" " -f1 | sort | uniq -c
      1 aaa@mail.ru
      2 bbb@mail.ru
      1 ccc@mail.ru
      1 ddd@mail.ru
[root@localhost ~]# _

```

Нетрудно заметить, что команда `uniq` рассматривает строки в целом, поэтому нужно обрезать нужные нам столбцы, чтобы обеспечить настоящее дублирование. В этом примере был выбран только первый столбец — email.

```
# cat file1.txt | cut -d" " -f1 | sort | uniq -c
```

выведет уникальные строки и подсчитает количество совпадений.

Команда `uniq` без параметров показала нам только уникальные записи. С параметром `-d` — только дубликаты. А с параметром `-c` — показала количестве совпадений строк. Может пригодится для ведения статистики.