

Основы системного администрирования и сетевых технологий

УРОК №7

Сортировка файлов с помощью команды `sort`

Существует большое количество утилит для сортировки, объединения, разбиения и прочей обработки текстовых файлов.

Команда `sort` выполняет сортировку входного потока по ключам сортировки. Она имеет следующий формат:
`sort [опции] [входные файлы]`

Команда выполняет конкатенацию указанных входных файлов, сортирует полученный текст и записывает результат в стандартный выходной поток. Если входные файлы не указаны, то ожидается ввод данных с клавиатуры.

Опции команды `sort` приведены в табл. 1.

Таблица 1. Опции команды sort

Опция	Описание
-c	Проверка того, отсортирован ли файл; сортировка не производится.
-m	Объединение отсортированных файлов; сортировка не производится.
-u	Удаление повторяющихся строк.
-o	Вывод результата не на экран, а в указанный файл.
-b	Игнорирование начальных пробелов в полях сортировки.
-n	Включение режима числовой сортировки.
-t	Задание разделителей полей.
-r	Сортировка в обратном порядке.
+поз1[,поз2]	Ключом сортировки становится строка, начинающаяся в позиции поз1 и заканчивающаяся перед позицией поз2 (или в конце текущей строки, если поз2 не указан); номера полей и позиции начальных символов отсчитываются от 0.
-k поз1[,поз2]	Ключом сортировки становится строка, начинающаяся в позиции поз1 и заканчивающаяся в позиции поз2 (или в конце текущей строки, если поз2 не указан); номера полей и позиции начальных символов отсчитываются от 1.
-p	Поле с номером p не должно сортироваться.

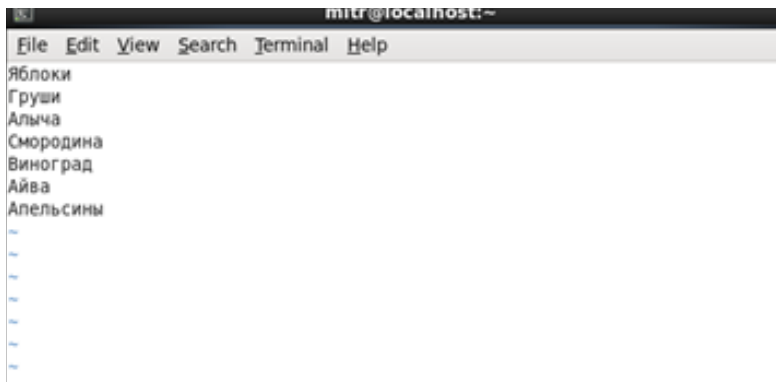
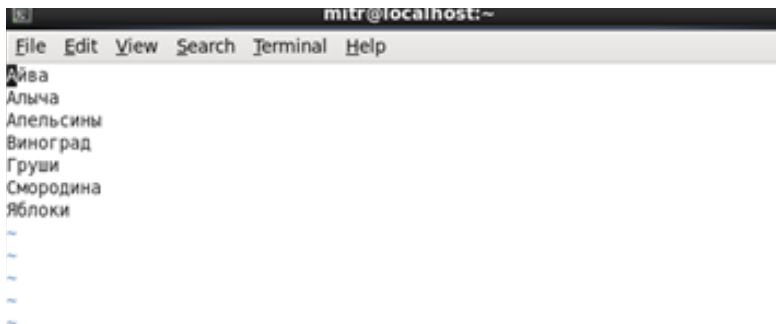


Рисунок 1. Файл a.txt в vi

Чтобы сохранить результаты сортировки используется опция `o` и указывается выходной файл. Можно также использовать переадресацию с помощью оператора `>`, например:
`sort a.txt > b.txt`

Создадим с помощью vi файл a.txt , показанный на рис. 1. vi a.txt



Выполним команду `sort a.txt > b.txt` и посмотрим результат ее выполнения с помощью vi. Файл b.txt в vi показан на рис. 2.

Рисунок 2. Файл b.txt в vi

```
mitr@localhost: ~  
File Edit View Search Terminal Help  
Яблоки:90  
Груши:110  
Алыча:100  
Смородина:200  
Виноград:120  
Айва:75  
Апельсины:60  
  
  
  
  

```

Рисунок 3. Измененный файл a.txt

```
mml@localhost:~$ cat /etc/passwd | grep -E "(apple|grape)"
```

```
File Edit View Search Terminal Help
```

```
Яблочко:100  
Груша:110  
Виноград:120  
Смородина:200  
Апельсины:60  
Айва:75  
Яблоки:90
```

Рисунок 4. Файл, отсортированный по второму полю

Для сортировки по определенным полям используется опция `-k`. Она указывается в формате:
`-k поз1, [поз2...]`, где поз1 и т.д. – номер позиции (столбца), по которому осуществляется сортировка.
 Изменим файл `a.txt` так, как показано на рис. 3.

Для его сортировки по второму столбцу можно выполнить следующую команду:

```
sort -t: -k 2 a.txt > c.txt
```

и посмотрим содержимое c.txt (рис. 4). Опция -t: используется для того, чтобы определить разделитель между полями. По умолчанию разделителем является пробел или символ табуляции.

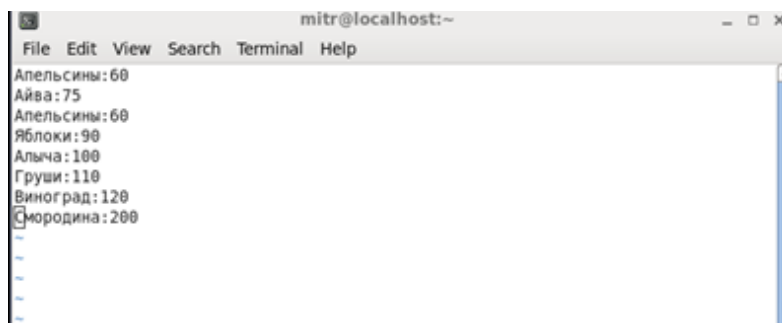


```
mitr@localhost:~  
File Edit View Search Terminal Help  
Апельсины: 60  
Айва: 75  
Яблоки: 90  
Алыча: 100  
Груши: 110  
Виноград: 120  
Смородина: 200  
~  
~  
~  
~  
~
```

Рисунок 5. Сортировка с числовым режимом

Обратите внимание на то, что сортировка по второму полю проведена, но числа сравниваются как строки: «100» < «60», поскольку «1» < «6». Теперь с помощью опции `-n` включим режим числовой сортировки.

```
sort -n -t: -k 2 a.txt > b.txt
```



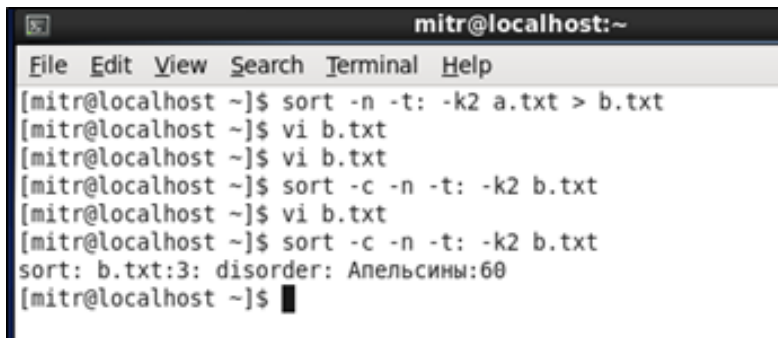
```
mitr@localhost:~  
File Edit View Search Terminal Help  
Апельсины: 60  
Айва: 75  
Апельсины: 60  
Яблоки: 90  
Алыча: 100  
Груши: 110  
Виноград: 120  
Смородина: 200  
~  
~  
~  
~  
~
```

Рисунок 5. Не отсортированный файл

Проверим теперь, является ли файл отсортированным. Сортировать не надо, надо просто проверить. Сделаем это, используя опцию `-c`:

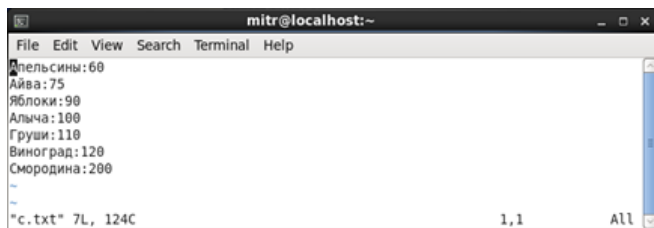
```
sort -c -n -t: -k 2 b.txt
```

Поскольку никаких сообщений в консоль не выведено, то файл отсортирован. Изменим `b.txt`, нарушив сортировку, как это показано на рис. 6.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort -n -t: -k2 a.txt > b.txt  
[mitr@localhost ~]$ vi b.txt  
[mitr@localhost ~]$ vi b.txt  
[mitr@localhost ~]$ sort -c -n -t: -k2 b.txt  
[mitr@localhost ~]$ vi b.txt  
[mitr@localhost ~]$ sort -c -n -t: -k2 b.txt  
sort: b.txt:3: disorder: Апельсины:60  
[mitr@localhost ~]$
```

Рисунок 6. Реакция команды на неотсортированный файл



```
mitr@localhost:~  
File Edit View Search Terminal Help  
Апельсины:60  
Айва:75  
Яблоки:90  
Алыча:100  
Груши:110  
Виноград:120  
Смородина:200  
~  
~  
"c.txt" 7L, 124C 1,1 All
```

Рисунок 7. Результат выполнения команды

Проверим теперь, является ли файл отсортированным. Сортировать не надо, надо просто проверить. Сделаем это, используя опцию `-c`:

`sort -c -n -t: -k 2 b.txt`

Поскольку никаких сообщений в консоль не выведено, то файл отсортирован. Изменим `b.txt`, нарушив сортировку, как это показано на рис. 6.

Опция `-u` удаляет из результатов дублирующие записи и выводит только уникальные поля.

Результат выполнения команды `sort -u -n -t: -k 2 b.txt > c.txt` показан на рисунке 7.

Команда `uniq`

Опция	Описание
<code>-u</code>	Отображение только не повторяющихся строк
<code>-d</code>	Отображение одной копии каждой повторяющейся строки
<code>-c</code>	Удаление повторяющихся строк с выводом перед каждой из оставшихся строк числа повторений
<code>-fn</code>	Игнорировать первые n полей; полем считается последовательность непробельных символов, завершающаяся пробелом или табуляцией

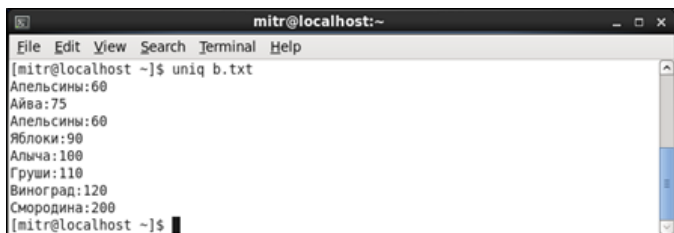
Таблица 2. Опции команды `uniq`

Команда `uniq` применяется для удаления идущих подряд повторяющихся строк. Эту команду можно рассматривать как вариант опции `-u` команды `sort`. Однако, `sort -u` позволяет избавиться от всех одинаковых строк в файле, тогда как `uniq` обнаруживает повторяющиеся строки только тогда, когда они следуют одна за другой.

Формат команды `uniq` следующий:
`uniq [Опции] ... [ВХОДНОЙ ФАЙЛ [ВЫХОДНОЙ ФАЙЛ]]`
 В табл. 2 приведены некоторые опции команды `uniq`.

Применим команду `uniq` к файлу `b.txt`, содержимое которого приведено на рис. 5.
`uniq b.txt`

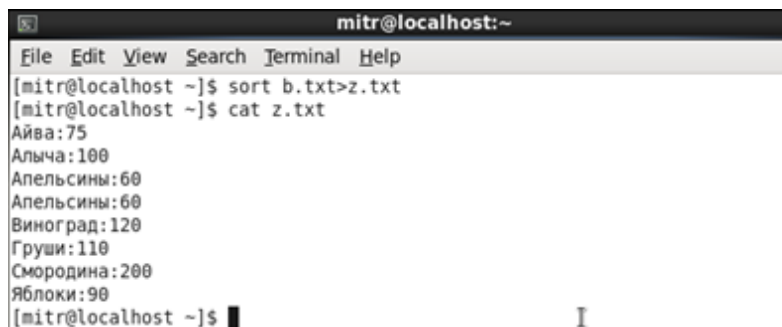
Результат выполнения команды показан на рис. 8.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ uniq b.txt  
Апельсины:60  
Айва:75  
Апельсины:60  
Яблоки:90  
Алыча:100  
Груши:110  
Виноград:120  
Смородина:200  
[mitr@localhost ~]$
```

Рисунок 8. Результат выполнения команды

Применим команду `uniq` к файлу `b.txt`, содержимое которого приведено на рис. 5.
`uniq b.txt`

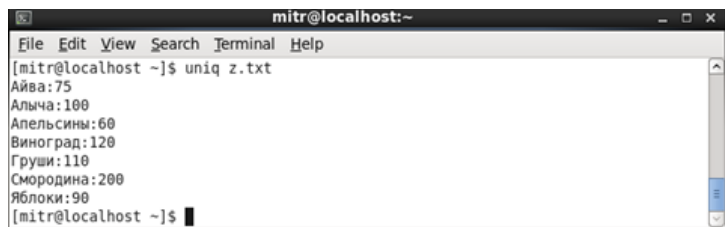


```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort b.txt > z.txt  
[mitr@localhost ~]$ cat z.txt  
Айва:75  
Алыча:100  
Апельсины:60  
Апельсины:60  
Виноград:120  
Груши:110  
Смородина:200  
Яблоки:90  
[mitr@localhost ~]$
```

Рисунок 9. Результат сортировки

Поскольку идущих подряд одинаковых строк нет, то вывод команды совпадает с содержимым файла.
Отсортируем файл `b.txt` с помощью команды `sort b.txt > z.txt`

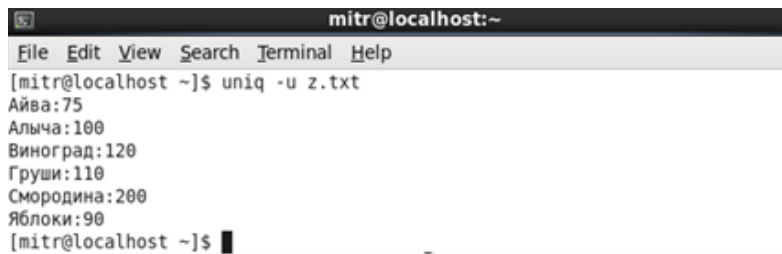
Результат сортировки показан на рис. 9.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ uniq z.txt  
Айва:75  
Алыча:100  
Апельсины:60  
Виноград:120  
Груши:110  
Смородина:200  
Яблоки:90  
[mitr@localhost ~]$
```

Рисунок 10. Результат выполнения команды

Применим команду `uniq z.txt`
ко вновь полученному файлу. Теперь в нем есть идущие подряд одинаковые строки. Результат выполнения команды показан на рис. 10.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ uniq -u z.txt  
Айва:75  
Алыча:100  
Виноград:120  
Груши:110  
Смородина:200  
Яблоки:90  
[mitr@localhost ~]$
```

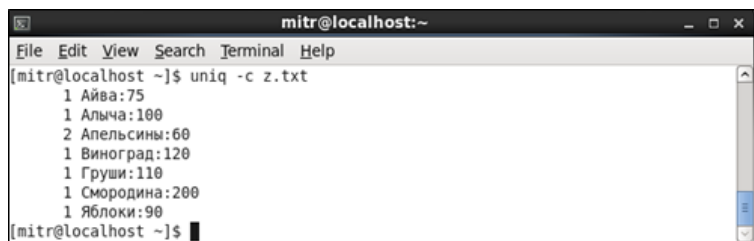
Рисунок 11. Результат выполнения команды

Воспользуемся ключом `-u`:

```
uniq -u z.txt
```

Результат выполнения команды показан на рис. 11.

Повторяющиеся идущие подряд строки не выведены.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ uniq -c z.txt  
1 Айва:75  
1 Алыча:100  
2 Апельсины:60  
1 Виноград:120  
1 Груши:110  
1 Смородина:200  
1 Яблоки:90  
[mitr@localhost ~]$
```

Рисунок 12. Результат выполнения команды

Воспользуемся ключом `-c`:

```
uniq -c z.txt
```

Результат выполнения команды показан на рис. 12.

Теперь возле каждой из идущих подряд повторяющихся строк указано количество повторений.

Команда join

Команда join выполняет соединение двух текстовых файлов на основании совпадения указанных полей. Ее действие напоминает операцию JOIN языка SQL. Команда работает следующим образом.

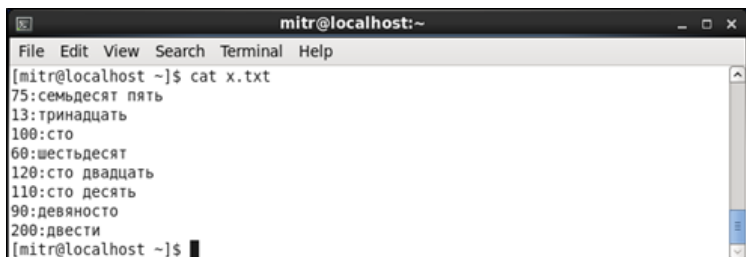
- 1) Каждый из двух входных файлов разбивается на поля, по умолчанию разделителем полей является пробел.
 - 2) Из первого файла извлекается первая строка, а из нее – указанное поля (по умолчанию – первое поле).
 - 3) Во втором файле ищется строка, имеющая такое же поле.
 - 4) Найденная строка, из которой удаляется найденное поле, присоединяется к исходной строке и результат записывается в выходной поток.
 - 5) Пункты 3 и 4 повторяются до тех пор, пока во втором файле есть строки с совпадающим полем.
 - 6) Пункты 2 – 5 повторяются для каждой строки первого файла. Таким образом, в выходной поток по умолчанию попадают только строки, имеющие общие компоненты.
- Важно: первый и второй файлы должны быть отсортированы по полям, по которым выполняется соединение.

Синтаксис команды:

join [опции] файл1 файл2

Наиболее часто используются следующие опции:

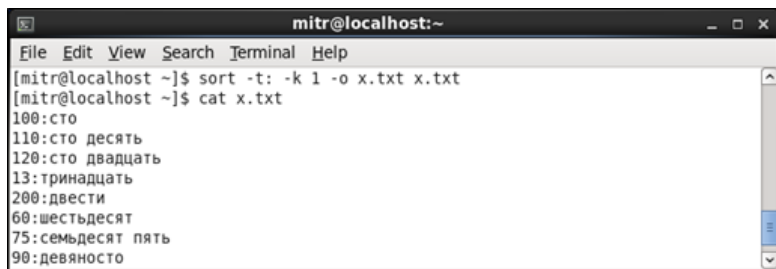
Опция	
-a n	Задаёт включение в выходной поток строк из файла n (n может быть 1 или 2), для которых не найдено ни одного совпадения по указанному полю.
-o формат	Определяет формат выводимой строки. Параметр «формат» представляет собой разделённый запятыми или пробелами список спецификаций, каждая из которых имеет формат номер_файла.поле. По умолчанию используется следующий формат: 1 – поле, по которому производится объединение; 2 – оставшаяся часть первой строки; 3 – оставшаяся часть второй строки.
-1 поле	Объединять строки по указанному полю первого файла (по умолчанию – первое поле)
-2 поле	Объединять строки по указанному полю второго файла (по умолчанию – первое поле)
-t	Задаёт разделитель полей во входном и выходном потоке



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ cat x.txt  
75:семьдесят пять  
13:тринадцать  
100:сто  
60:шестьдесят  
120:сто двадцать  
110:сто десять  
90:девяносто  
200:двести  
[mitr@localhost ~]$
```

Рисунок 13. Файл x.txt

Пусть есть файл z.txt, приведенный на рис. 9. Создадим файл x.txt, приведенный на рис. 13.

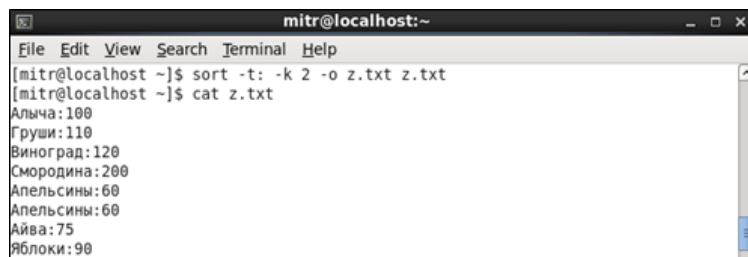


```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort -t: -k 1 -o x.txt x.txt  
[mitr@localhost ~]$ cat x.txt  
100:сто  
110:сто десять  
120:сто двадцать  
13:тринадцать  
200:двести  
60:шестьдесят  
75:семьдесят пять  
90:девяносто
```

Рисунок 14. Отсортированный по первому полю файл x.txt

Пусть необходимо выполнить соединение по числовому полю, второму в первом файле и первому во втором файле.

Отсортируем файл x.txt по первому полю с помощью команды `sort`, как это показано на рис. 14.

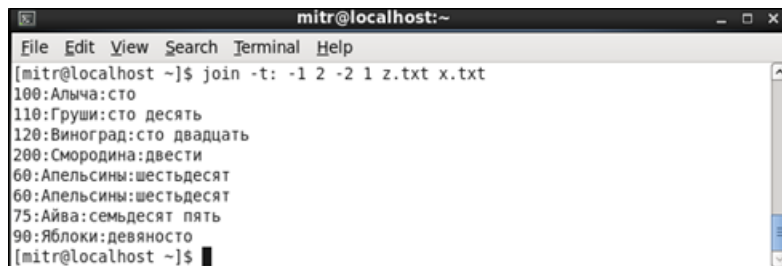


```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort -t: -k 2 -o z.txt z.txt  
[mitr@localhost ~]$ cat z.txt  
Алыча:100  
Груши:110  
Виноград:120  
Смородина:200  
Апельсины:60  
Апельсины:60  
Айва:75  
Яблоки:90
```

Обратите внимание на то, что сортировка чисел проводится по правилам сортировки строк, поскольку при сортировке не использована опция `-n` и, следовательно, режим числовой сортировки не включен.

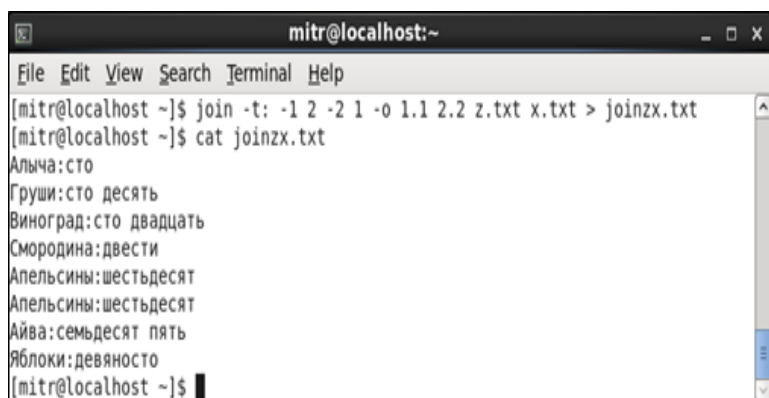
Отсортируем файл z.txt по второму полю, как это показано на рис. 15.

Рисунок 15. Отсортированный файл z.txt



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ join -t: -1 2 -2 1 z.txt x.txt  
100:Алыча:сто  
110:Груши:сто десять  
120:Виноград:сто двадцать  
200:Смородина:двести  
60:Апельсины:шестьдесят  
60:Апельсины:шестьдесят  
75:Айва:семьдесят пять  
90:Яблоки:девятьсто  
[mitr@localhost ~]$
```

Рисунок 16. Присоединение файла x.txt к файлу z.txt



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ join -t: -1 2 -2 1 -o 1.1 2.2 z.txt x.txt > joinzx.txt  
[mitr@localhost ~]$ cat joinzx.txt  
Алыча:сто  
Груши:сто десять  
Виноград:сто двадцать  
Смородина:двести  
Апельсины:шестьдесят  
Апельсины:шестьдесят  
Айва:семьдесят пять  
Яблоки:девятьсто  
[mitr@localhost ~]$
```

Рисунок 17. Присоединение файла с нужными полями

Выполним присоединение файла x.txt к файлу z.txt, как это показано на рис. 16.

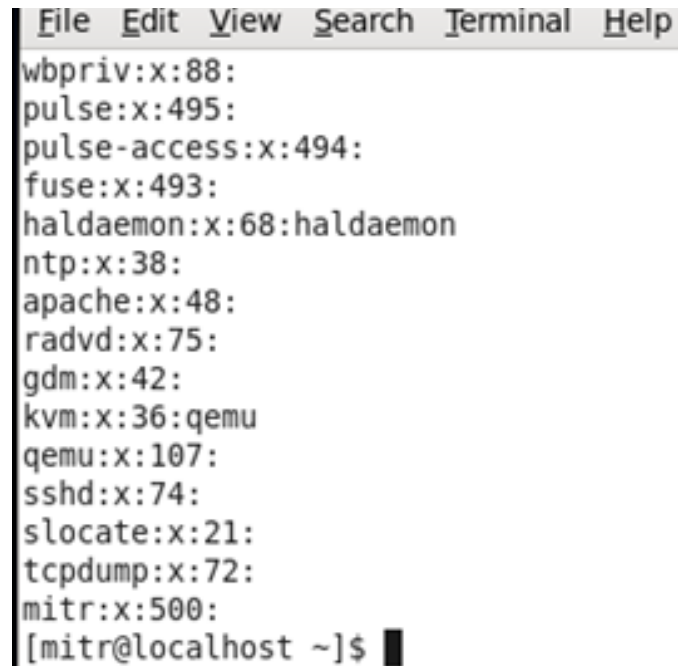
Присоединение произошло. Если необходимо оставить в выводе только второе и третье поле и сохранить результаты в файле, то воспользуемся командой join с ключами, показанными на рис. 17.

Теперь скопируем в домашний каталог файлы /etc/passwd и /etc/group с помощью команд
cp /etc/passwd password.txt
cp /etc/group group.txt
Содержимое файлов password.txt и group.txt показано на рис. 18 и рис. 19.

```
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
saslauth:x:498:76:Saslauthd user:/var/empty/saslauth:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
pulse:x:497:495:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
radvd:x:75:75:radvd user:/:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
qemu:x:107:107:qemu user:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
mitr:x:500:500:mitr:/home/mitr:/bin/bash
[mitr@localhost ~]$
```

Рисунок 18. Содержимое файла password.txt (сокращено)

Рисунок 19.
Содержимое файла
group.txt (сокращено)



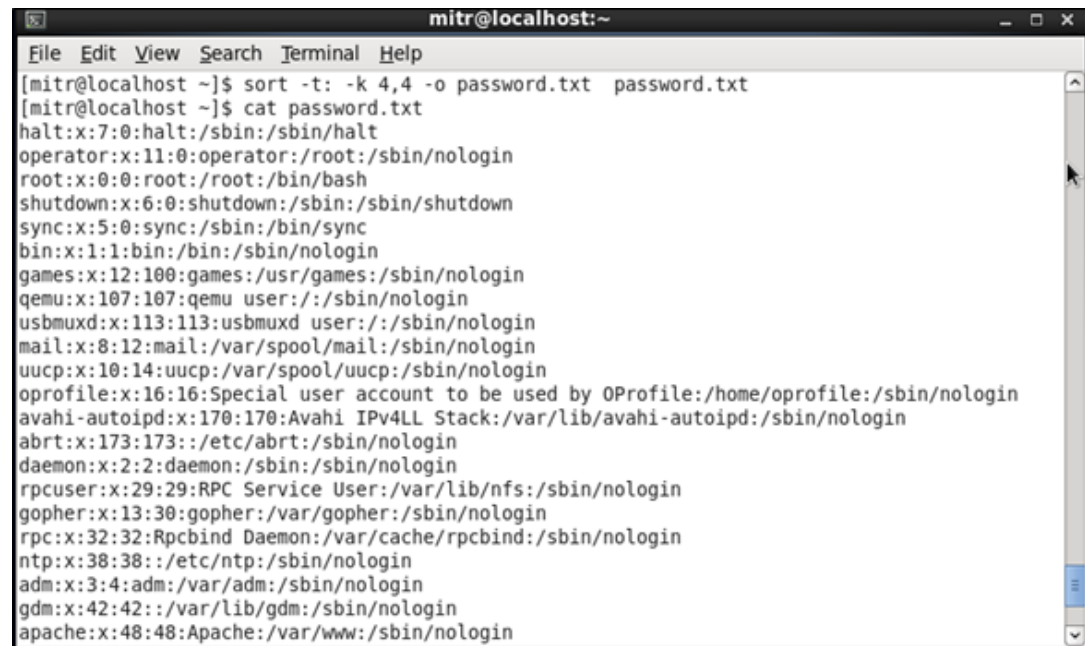
```
File Edit View Search Terminal Help
wbpriv:x:88:
pulse:x:495:
pulse-access:x:494:
fuse:x:493:
haldaemon:x:68:haldaemon
ntp:x:38:
apache:x:48:
radvd:x:75:
gdm:x:42:
kvm:x:36:qemu
qemu:x:107:
sshd:x:74:
slocate:x:21:
tcpdump:x:72:
mitr:x:500:
[mitr@localhost ~]$
```


Соединим эти файлы, чтобы получить в одном файле входное имя пользователя, имя группы (не GID, а имя группы) и домашний каталог пользователя.

В этих файлах есть общее поле – идентификатор группы (GID). В файле password.txt оно находится в четвертом поле, в файле group.txt – в третьем поле. Входное имя пользователя находится в первом поле password.txt, имя группы – в первом поле файла group.txt, домашний каталог – в шестом поле файла password.txt.

Отсортируем password.txt по четвертому полю (рис 20), а group.txt – по третьему, как это показано на рис. 21.

Рисунок 20.
Файл password.txt, отсортированный
по четвертому полю (сокращено)



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort -t: -k 4,4 -o password.txt password.txt  
[mitr@localhost ~]$ cat password.txt  
halt:x:7:0:halt:/sbin:/sbin/halt  
operator:x:11:0:operator:/root:/sbin/nologin  
root:x:0:0:root:/root:/bin/bash  
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown  
sync:x:5:0:sync:/sbin:/bin/sync  
bin:x:1:1:bin:/bin:/sbin/nologin  
games:x:12:100:games:/usr/games:/sbin/nologin  
qemu:x:107:107:qemu user:/:/sbin/nologin  
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin  
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin  
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin  
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin  
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin  
abrt:x:173:173:./etc/abrt:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin  
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin  
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin  
ntp:x:38:38:./etc/ntp:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin  
gdm:x:42:42:./var/lib/gdm:/sbin/nologin  
apache:x:48:48:Apache:/var/www:/sbin/nologin
```


Рисунок 21.
Файл group.txt, отсортированный
по третьему полю (сокращено)



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ sort -t: -k 3,3 -o group.txt group.txt  
[mitr@localhost ~]$ cat group.txt  
root:x:0:  
bin:x:1:bin,daemon,mitr  
wheel:x:10:  
users:x:100:  
qemu:x:107:  
cdrom:x:11:  
usbmuxd:x:113:  
mail:x:12:mail,postfix  
uucp:x:14:  
man:x:15:  
stapusr:x:156:  
stapusr:x:157:  
stapdev:x:158:  
oprofile:x:16:  
avahi-autoipd:x:170:  
abrt:x:173:  
dialout:x:18:  
floppy:x:19:  
daemon:x:2:bin,daemon  
games:x:20:  
slocate:x:21:  
utmp:x:22:
```

Рисунок 22.
Соединение файлов
password.txt и group.txt
по полю GID (сокращено)

Теперь соединим файлы password.txt и group.txt по полю GID и выведем необходимые поля с помощью опции -o, как это показано на рис. 22.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ join -t: -1 4 -2 3 -o 1.1 2.1 1.6 password.txt group.txt  
halt:root:/sbin  
operator:root:/root  
root:root:/root  
shutdown:root:/sbin  
sync:root:/sbin  
bin:bin:/bin  
games:users:/usr/games  
qemu:qemu:/  
usbmuxd:usbmuxd:/  
mail:mail:/var/spool/mail  
uucp:uucp:/var/spool/uucp  
oprofile:oprofile:/home/oprofile  
avahi-autoipd:avahi-autoipd:/var/lib/avahi-autoipd  
abrt:abrt:/etc/abrt  
daemon:daemon:/sbin  
rpcuser:rpcuser:/var/lib/nfs  
gopher:gopher:/var/gopher  
rpc:rpc:/var/cache/rpcbind  
ntp:ntp:/etc/ntp  
adm:adm:/var/adm  
gdm:gdm:/var/lib/gdm  
apache:apache:/var/www  
pulse:pulse:/var/run/pulse
```

Команда cut

Опция	Описание
-с список	Определяет, какие символы извлекаются из каждого входного файла
-f список	Определяет, какие поля извлекаются из каждого входного файла
-d	Задаёт разделитель полей

Команда cut предназначена для вырезания (выделения) фрагментов строк из текстовых файлов или входного потока. Выделенный текст направляется в выходной поток. Синтаксис команды:

cut OPTION... [FILE]...

Часто используемые опции команды представлены в таблице.

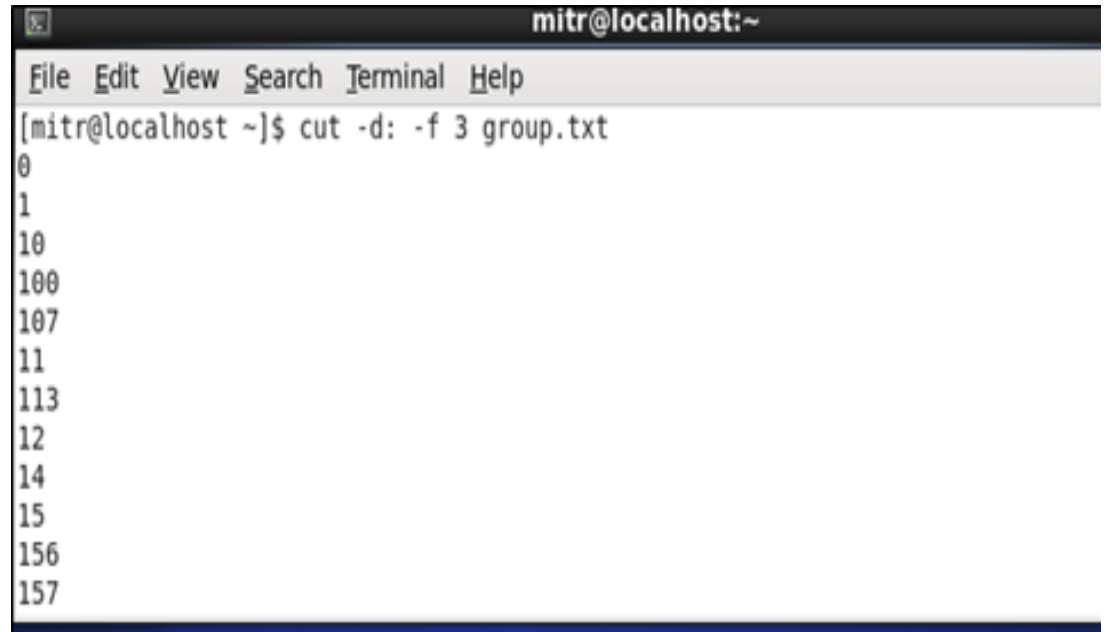
Параметр список в опциях -с и -f представляет собой диапазон символов или полей, разделённых запятыми.

Список может быть описан в одной из следующих форм:

- n – в выходной поток включается каждый n-ый символ или поле каждой строки входного файла;
- n- – диапазон формируется от n-ого символа до окончания строки;
- n-m – диапазон от n-ого символа (или поля) включительно до m-ого символа (или поля) включительно;
- -m – диапазон от начала строки до m-ого символа включительно.

Выделим из файла group.txt третье поле, как это показано на рис. 23.

Рисунок 23.
Третье поле файла group.txt



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ cut -d: -f 3 group.txt  
0  
1  
10  
100  
107  
11  
113  
12  
14  
15  
156  
157
```

Команда paste

Команда paste объединяет строки с одинаковыми номерами указанных файлов и возвращает объединенные строки в выходной поток.

Синтаксис команды:

cut OPTION... [FILE]...

Часто используемые опции команды приведены в таблице.

Опция	Описание
-d список	По умолчанию объединенные строки отделяются символом табуляции. Опция позволяет определить другой разделитель или список разделителей. В случае использования списка, его элементы используются циклически, то есть первый разделитель разделяет первое и второе поле, второй – второе и третье и т.д. Когда символы в списке завершаются, при разделении очередных полей снова используется первый символ.
-s	Определяет режим последовательного слияния строк каждого входного файла по отдельности.
-	Осуществляет выборку строки из входного потока.

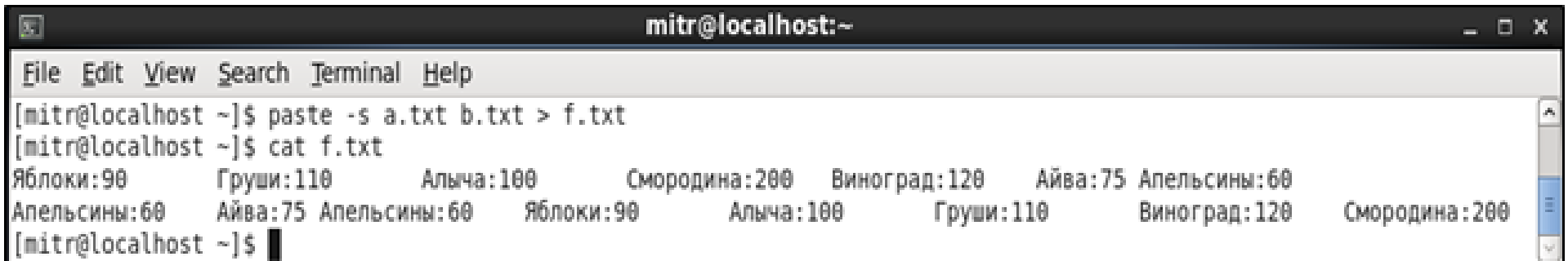
```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ paste group.txt group.txt  
root:x:0:      root:x:0:  
bin:x:1:bin,daemon,mitr bin:x:1:bin,daemon,mitr  
wheel:x:10:    wheel:x:10:  
users:x:100:   users:x:100:  
qemu:x:107:    qemu:x:107:  
cdrom:x:11:    cdrom:x:11:  
usbmuxd:x:113: usbmuxd:x:113:  
mail:x:12:mail,postfix mail:x:12:mail,postfix  
uucp:x:14:     uucp:x:14:  
man:x:15:     man:x:15:  
stapusr:x:156: stapusr:x:156:  
stapsys:x:157: stapsys:x:157:
```

На рис. 24 показан результат объединения файла group.txt с самим собой.

```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ paste -d= a.txt b.txt  
Яблоки:90=Апельсины:60  
Груши:110=Айва:75  
Алыча:100=Апельсины:60  
Смородина:200=Яблоки:90  
Виноград:120=Алыча:100  
Айва:75=Груши:110  
Апельсины:60=Виноград:120  
=Смородина:200  
[mitr@localhost ~]$
```

На рис. 25 показано использование опции -d.

При использовании опции `-s` команда `paste` выполняет слияние всех строк сначала первого файла, а затем второго. Таким образом, формируются столбцы из соответствующих по номерам строк каждого файла. Использование опции `-s` показано на рис. 27.



```
mitr@localhost:~  
File Edit View Search Terminal Help  
[mitr@localhost ~]$ paste -s a.txt b.txt > f.txt  
[mitr@localhost ~]$ cat f.txt  
Яблоки:90      Груши:110      Алыча:100      Смородина:200  Виноград:120  Айва:75  Апельсины:60  
Апельсины:60  Айва:75  Апельсины:60  Яблоки:90      Алыча:100      Груши:110      Виноград:120  Смородина:200  
[mitr@localhost ~]$
```

Рисунок 27. Использование опции `-s` команды `paste`

Команда **split**

Команда **split** предназначена для разделения больших файлов на более мелкие файлы. Команда **split** имеет следующий формат:

split [-размер_выходного_файла] входной_файл [префикс]
Первый параметр определяет количество строк, на которое необходимо разбить файл.

По умолчанию файл разбивается на фрагменты размером 1000 строк. Если размер разбиваемого файла не кратен 1000 строк, то последний файл будет содержать менее 1000 строк. Например, из файла, содержащего 3100 строк, получатся три файла по 1000 строк в каждом и один файл из 100 строк. Имя каждого созданного файла представляется в формате от префикс[aa] до префикс[zz]. По умолчанию префиксом является символ «x». Таким образом, команда **split** создает следующую последовательность файлов:

хаа, хаб, ... хзу, хzz.

Если расположить файлы в алфавитном порядке и выполнить их последовательную конкатенацию, то получится исходный файл.

Основы системного администрирования и сетевых технологий
