



[Главная](#) >> [Команды](#) >> Команда cut Linux

Команда cut Linux

Обновлено: 13 апреля 2020 Опубликовано: 20 декабря, 2019 от [ellado](#), 10 комментариев, время чтения: 13 минут

Обнаружили ошибку в тексте? Сообщите мне об этом. Выделите текст с ошибкой и нажмите **Ctrl+Enter**.

Команда `cut` используется, если нужно вырезать часть текста – при этом он может находиться в файле либо быть напечатанным через стандартный ввод. В Unix-системах эта команда удаляет секции текста, которые были обозначены при помощи байтов, символов или полей, разделенных знаками `"-"` и `:`. Работу `cut` обеспечивает одноименная утилита. Она входит в состав `GNU Coreutils`, так что по умолчанию доступна во всех дистрибутивах `Linux`. Также ее можно воспользоваться на машинах с операционной системой `BSD`.

Мы постараемся подробно рассказать о том, как работать с командой `cut`, задействовав и стандартный ввод, и текстовые файлы. Ну и, как всегда, дадим примеры использования, чтобы тем, кто впервые сталкивается с выполнением этой команды, было на что ориентироваться.

Содержание статьи

- [Синтаксис команды cut](#)
- [Параметры cut](#)

[Конфиденциальность](#) ·
[Условия использования](#)

Privacy

- [Примеры использования cut в Linux](#)
 - [1. Работа с байтами](#)
 - [2. Работа с символами](#)
 - [3. Работа со столбцами](#)
- [Выводы](#)

Синтаксис команды cut

Написание команды `cut` выглядит следующим образом:

```
$ cut опции путь_к_файлу
```

Использовать файл не обязательно. Если на месте его названия поставить прочерк - либо не указать ничего, команда возьмет текст из стандартного ввода. При необходимости можно указывать больше файлов, чем один.

Параметры cut

Список опций, которые позволяют управлять поведением команды:

- **-b (--bytes=LIST)** – номер байта, набор или диапазон байтов, подлежащих вырезанию.
- **-c (--characters=LIST)** – символ, который следует вырезать. Также можно указывать набор либо диапазон символов.
- **-d (--delimiter=DELIM)** – с помощью этой опции пользователь устанавливает свой разделитель вместо стандартного TAB.
- **-f (--fields=LIST)** – перечень полей для вырезания.
- **-s (--only-delimited)** – если была применена эта опция, `cut` не выводит строки, где нет разделителя.
- **--complement** – задает байты, символы или поля, которые останутся в файле или тексте из стандартного ввода. Все остальное будет вырезано.
- **--output-delimiter=STRING** – по умолчанию выходной разделитель соответствует входному. Эта опция позволяет задать другой выходной разделитель.
- **-z, --zero-terminated** – вместо символа новой строки разделителем будет NULL.

Это основные параметры команды `cut`, которые нам понадобятся при работе с утилитой.

Примеры использования cut в Linux

Прежде всего создадим файл `cut_command_example.txt` и поместим его в домашнюю директорию. В теле документа пропишем текст:

Privacy

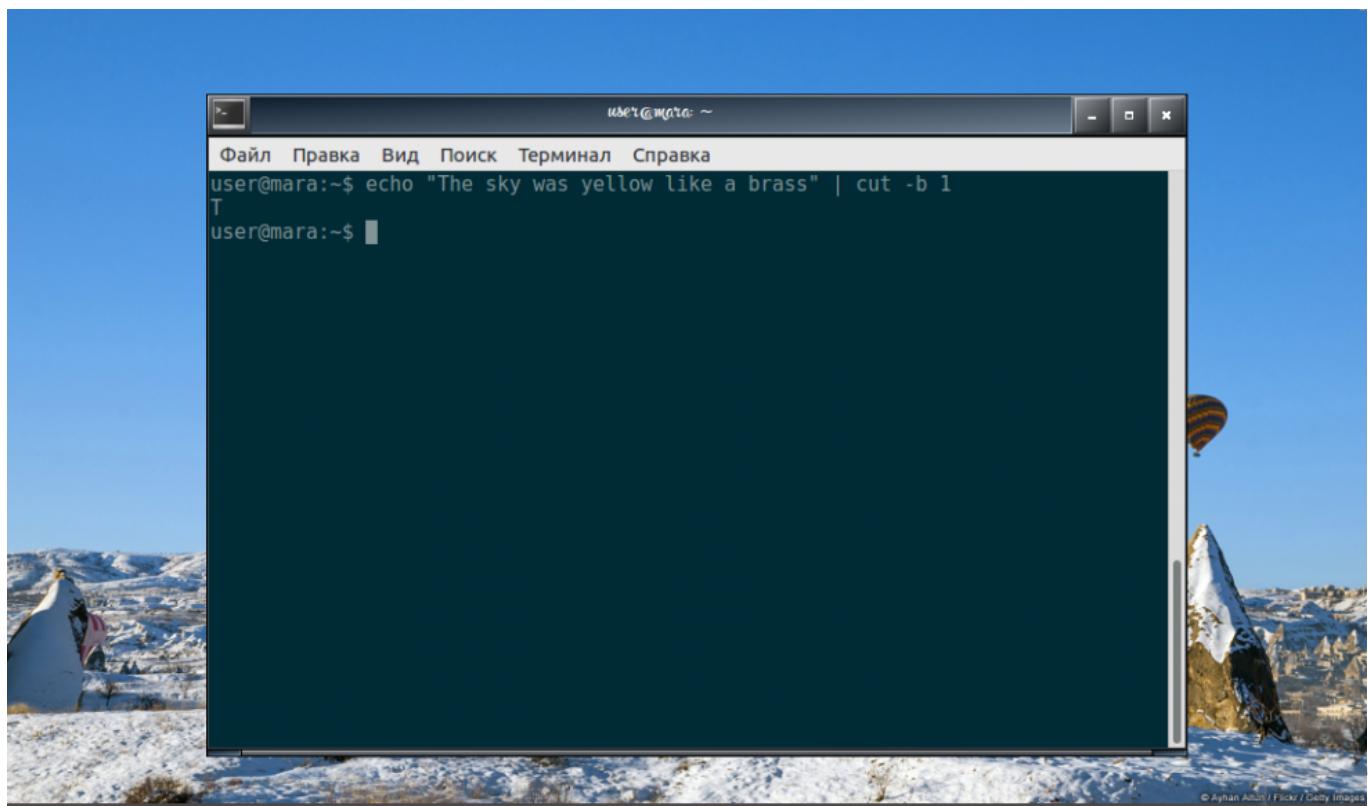
```
Winter: white: snow: frost
Spring: green: grass: warm
Summer: colorful: blossom: hot
Autumn: yellow: leaves: cool
```

1. Работа с байтами

На первый взгляд кажется, что с помощью байтов удобнее всего задавать координаты символов, которые нужно вырезать. Но не все так просто. В текстах, написанных на английском языке, каждый символ – это один байт, а вот в других языках некоторые буквы, цифры и спецсимволы могут занимать и два, и три, и четыре байта. Эту особенность следует учитывать, принимая во внимание систему кодировки текста, иначе результат выполнения команды будет не таким, как ожидалось.

Чтобы вырезать из текста, напечатанного через стандартный ввод, первый символ, команда `cut` должна иметь вид:

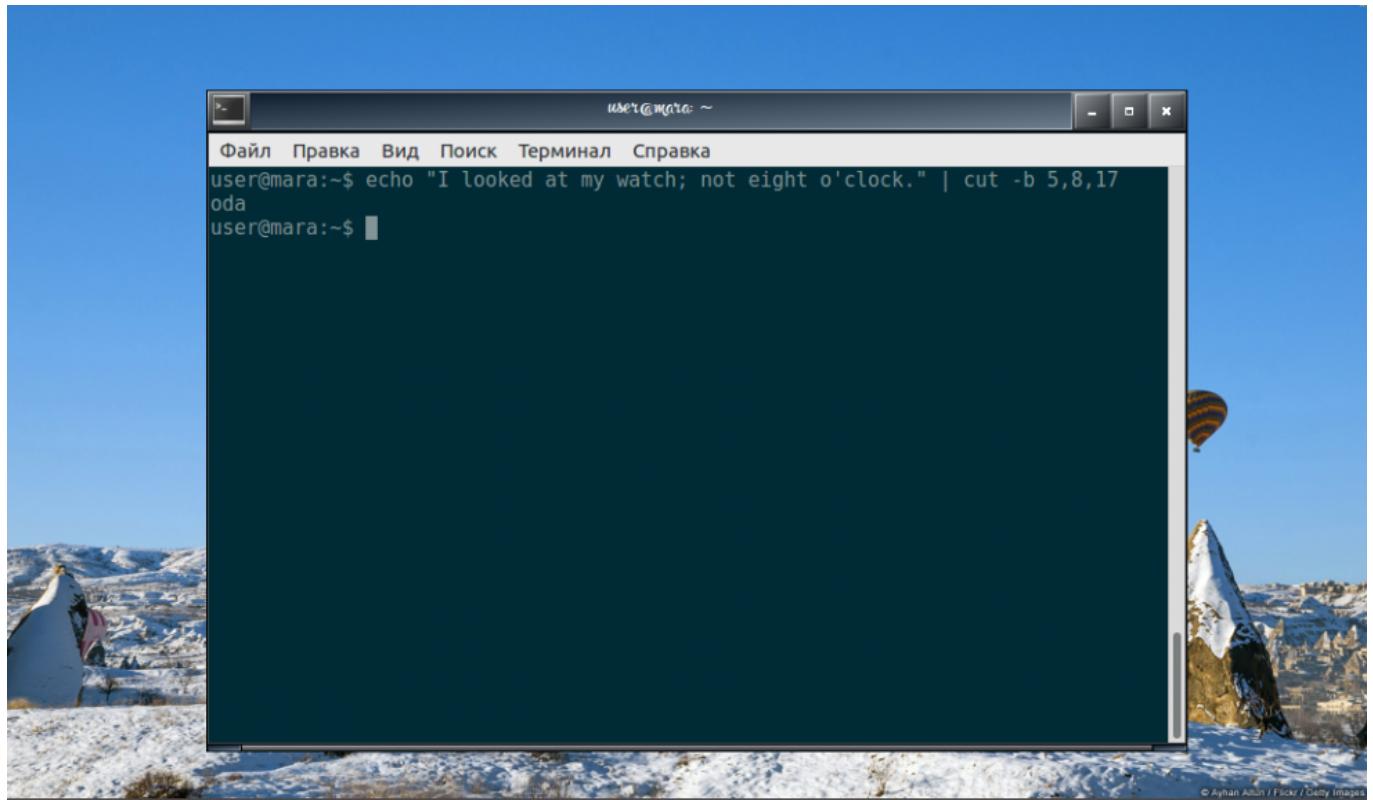
```
$ echo "The sky was yellow as brass." | cut -b 1
```



Если же необходимо вырезать несколько символов, их номера указывают через запятую:

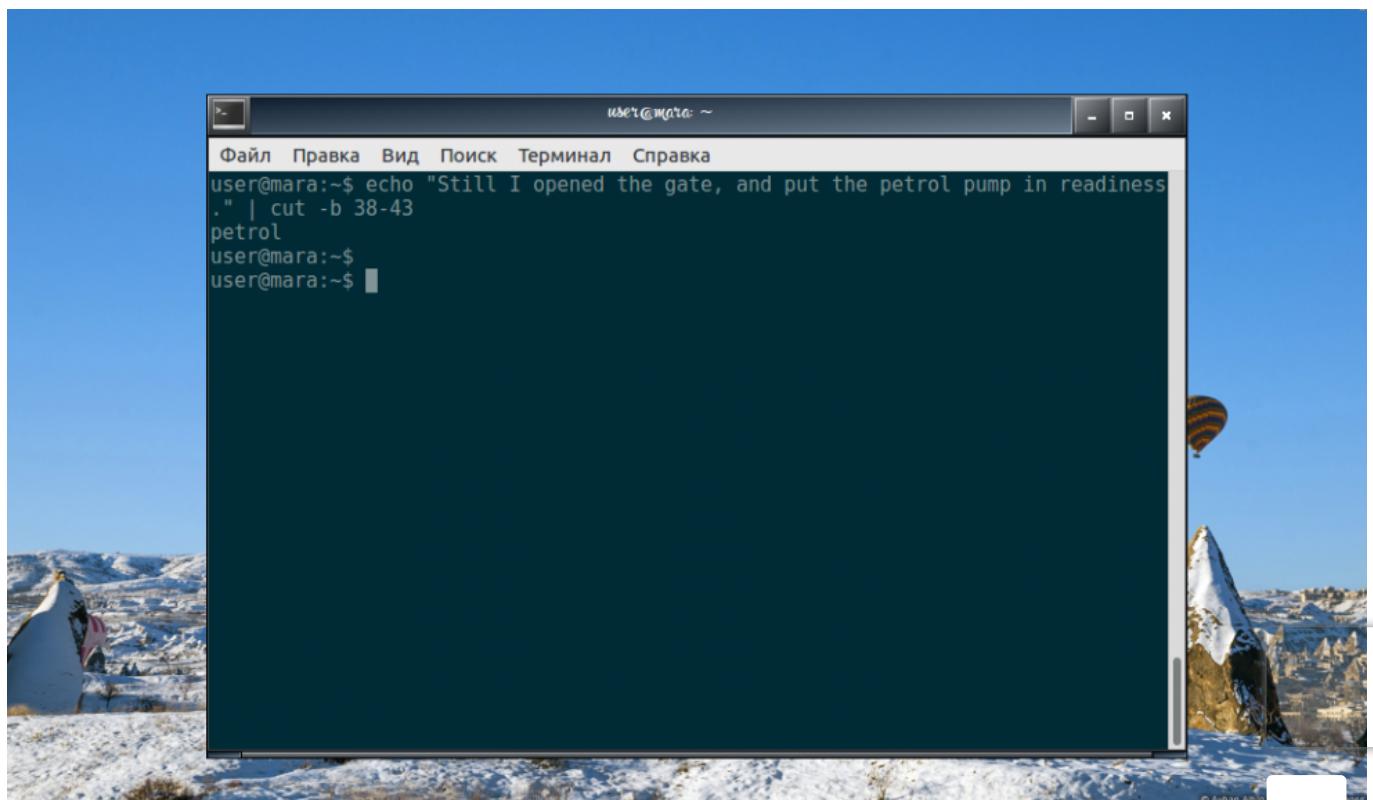
```
$ echo "I looked at my watch; not eight o'clock." | cut -b 5,8,17
```

Privacy



Нет ничего сложного и в том, чтобы вырезать некоторый диапазон символов. В этом случае номера символов указывают через дефис:

```
$ echo "Still I opened the gate, and put the petrol pump in readiness." | cut -b  
38-43
```

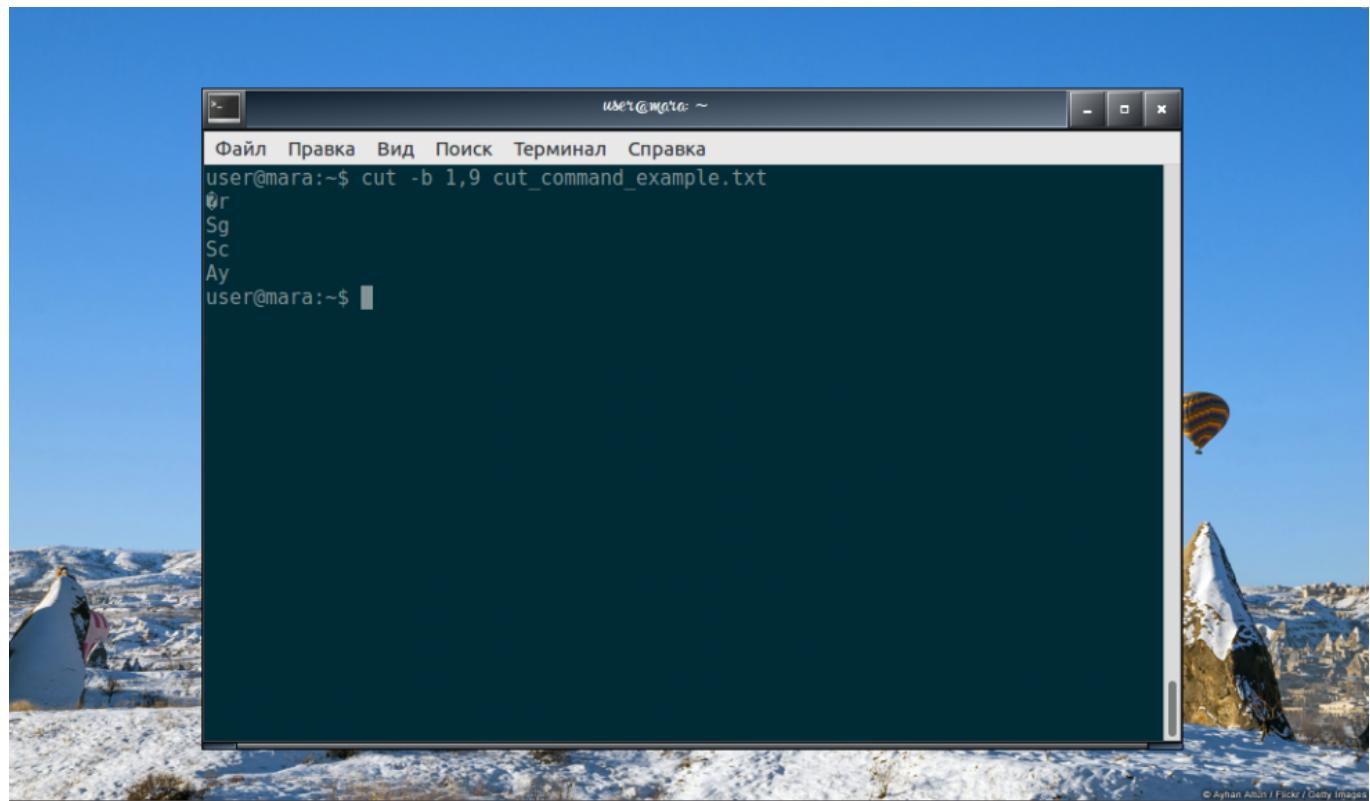


Когда в заданном диапазоне числовые значения превышают количество байт, содержащееся в тексте, на экран будет выведено сообщение об ошибке либо просто пустая строка (зависит от версии cut, которая установлена на компьютере).

Теперь рассмотрим те варианты использования команды, где нужно вырезать символы из текста, который находится в файле. Стандартная запись в терминале выглядит так:

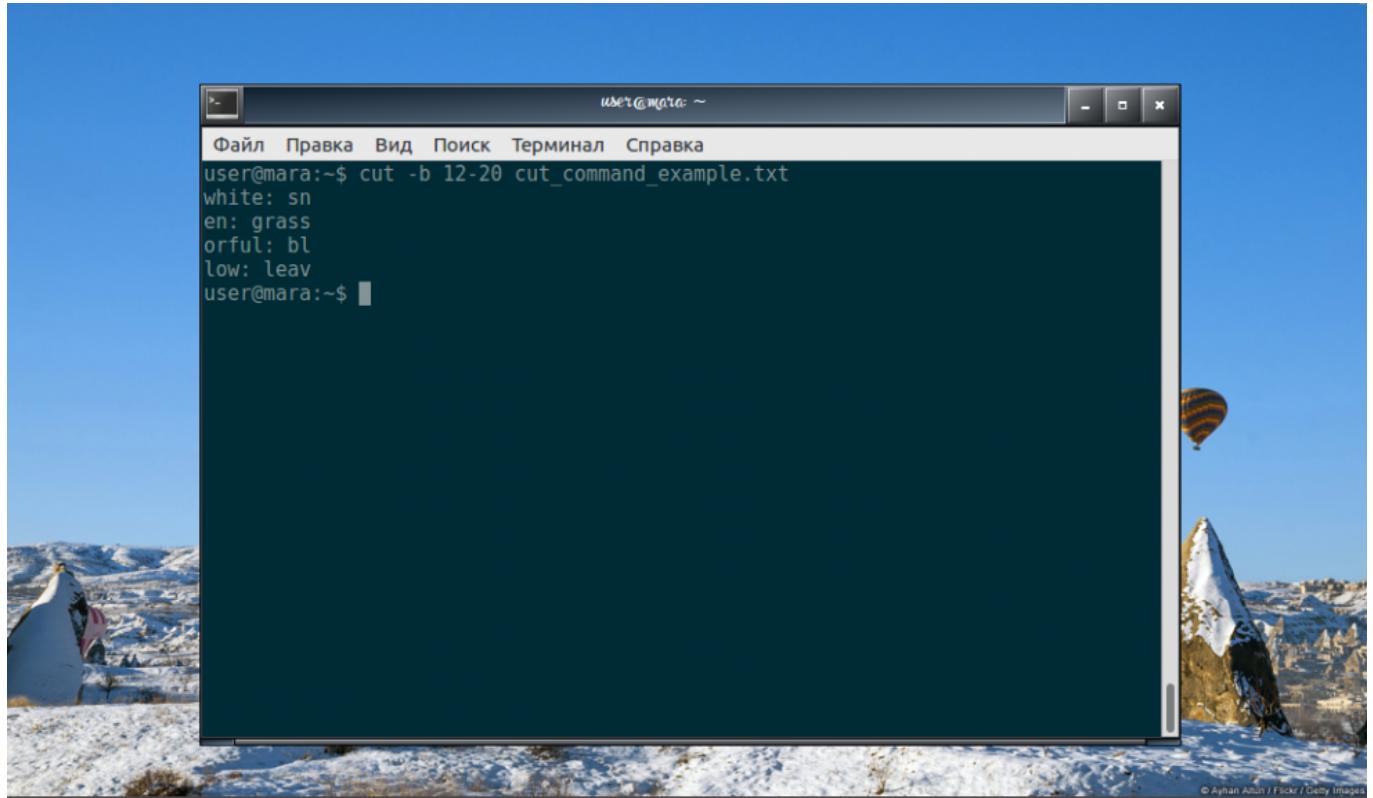
```
$ cut -b 1,9 cut_command_example.txt
```

Как вы можете помнить, в созданном ранее документе было четыре строки. Команда в каждой из них вырезала символы, соответствующие указанным номерам байтов, и вывела их также построчно.



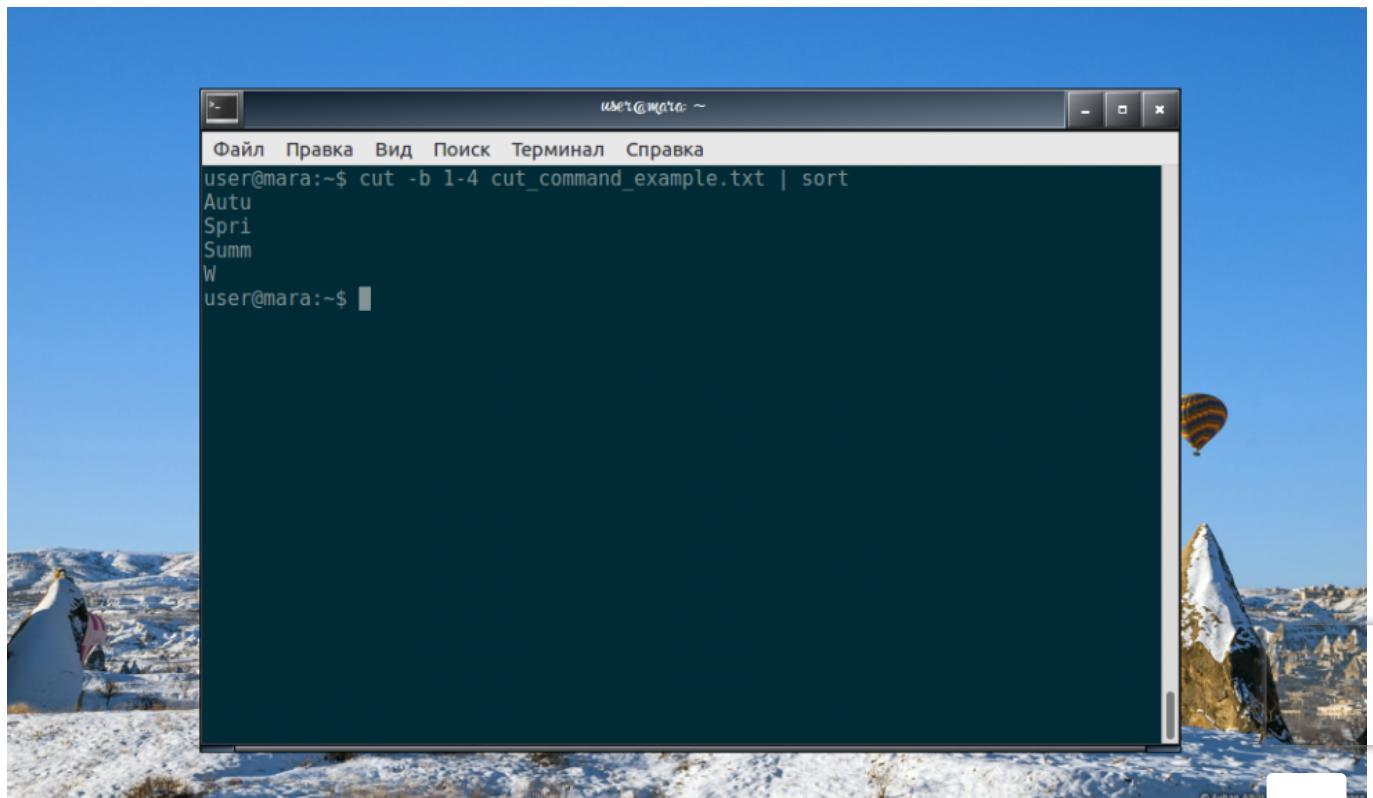
Примерно так же работает вырезание символов в заданном диапазоне. Диапазон - это два числа, написанные через дефис:

```
$ cut -b 12-20 cut_command_example.txt
```



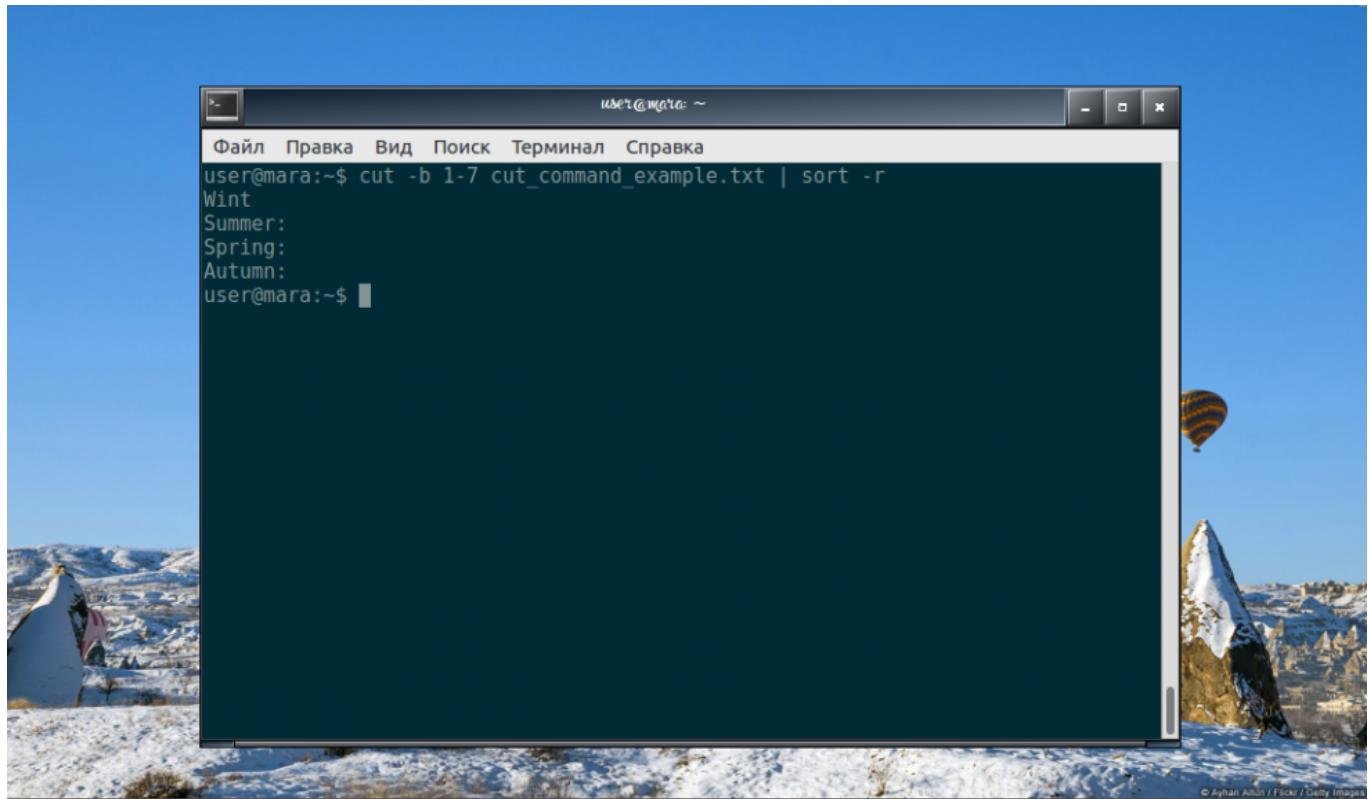
Команду `cut` можно использовать саму по себе, но не возбраняется сочетать с другими командами. Чаще всего используется `sort`. Попробуем вырезать первые 4 символа и расположить строки в алфавитном порядке:

```
$ cut -b 1-7 cut_command_example.txt | sort
```



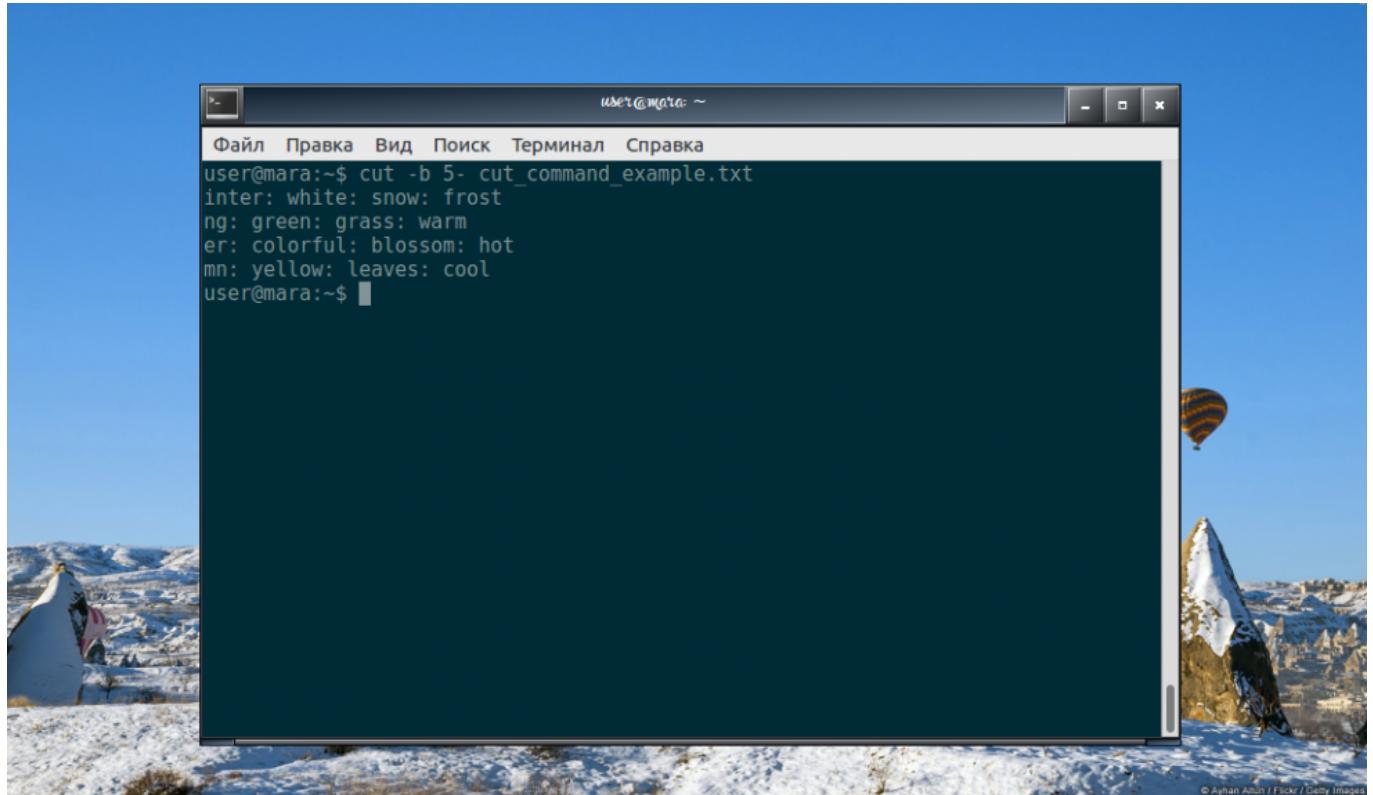
Для сортировки по алфавиту команде `sort` не нужны дополнительные опции. А вот чтобы расположить строки в обратном порядке, следует добавить к записи `-r`:

```
$ cut -b 1-7 cut_command_example.txt | sort -r
```



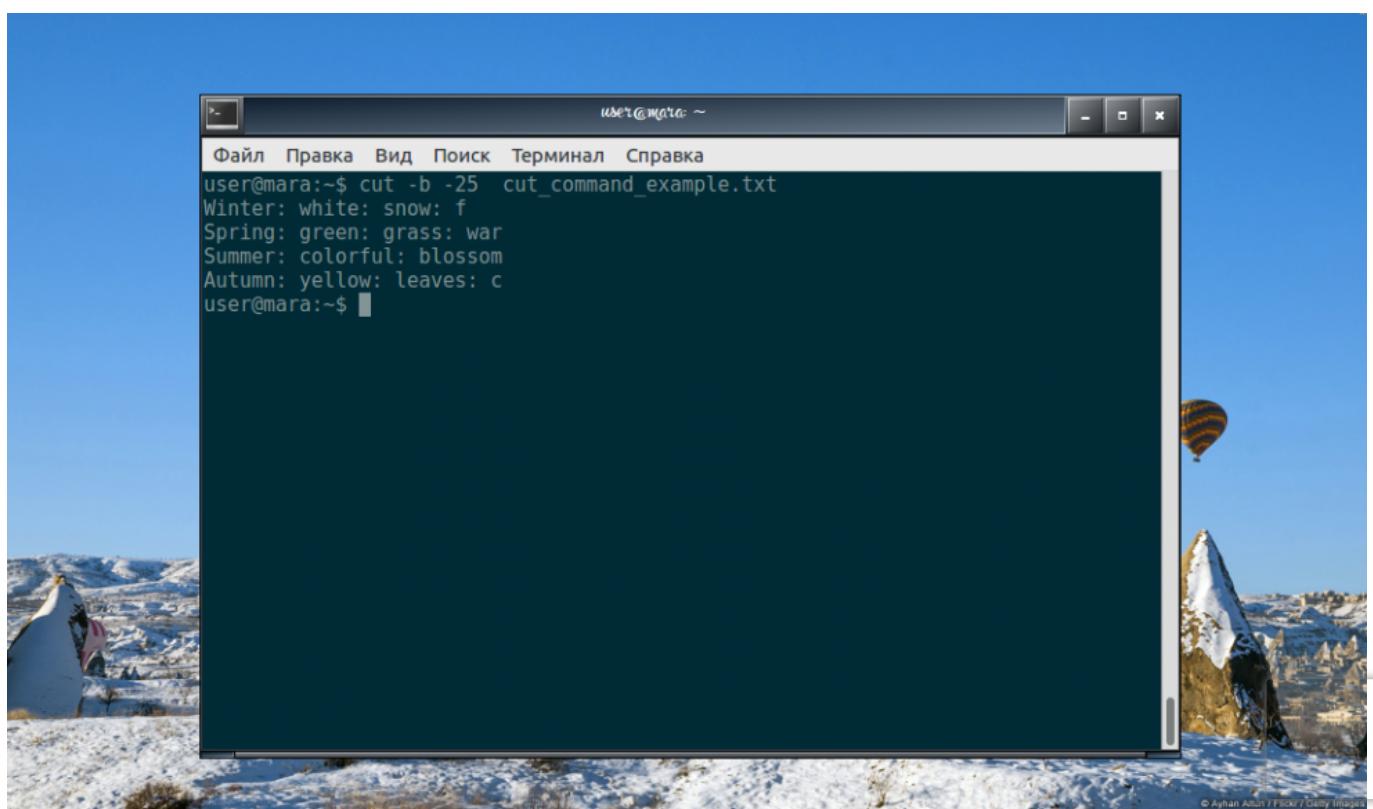
При помощи дефиса, который стоит рядом с числом, обозначающим номер байта, можно задавать неопределенные диапазоны символов. Например, чтобы команда вырезала все символы, начиная с пятого и заканчивая последним, нужно поставить дефис справа от числа:

```
$ cut -b 5- cut_command_example.txt
```



А чтобы были вырезаны все символы, начиная с первого байта и заканчивая шестнадцатым, дефис должен находиться справа от числа:

```
$ cut -b -25 cut_command_example.txt
```

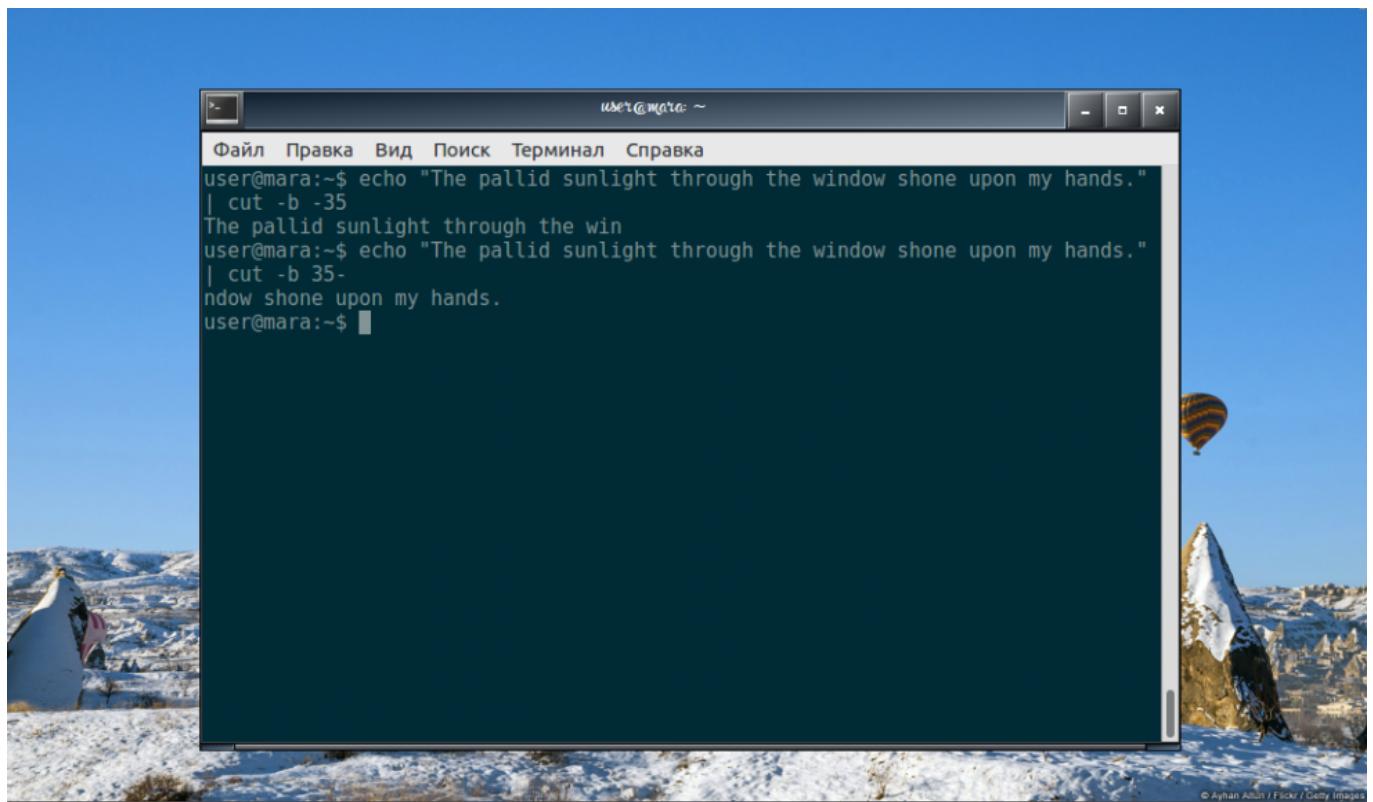


Точно так же дефис используется, если текст задан не из файла, а через стандарт ввод:

Privacy

```
$ echo "The pallid sunlight through the window shone upon my hands." | cut -b -35
```

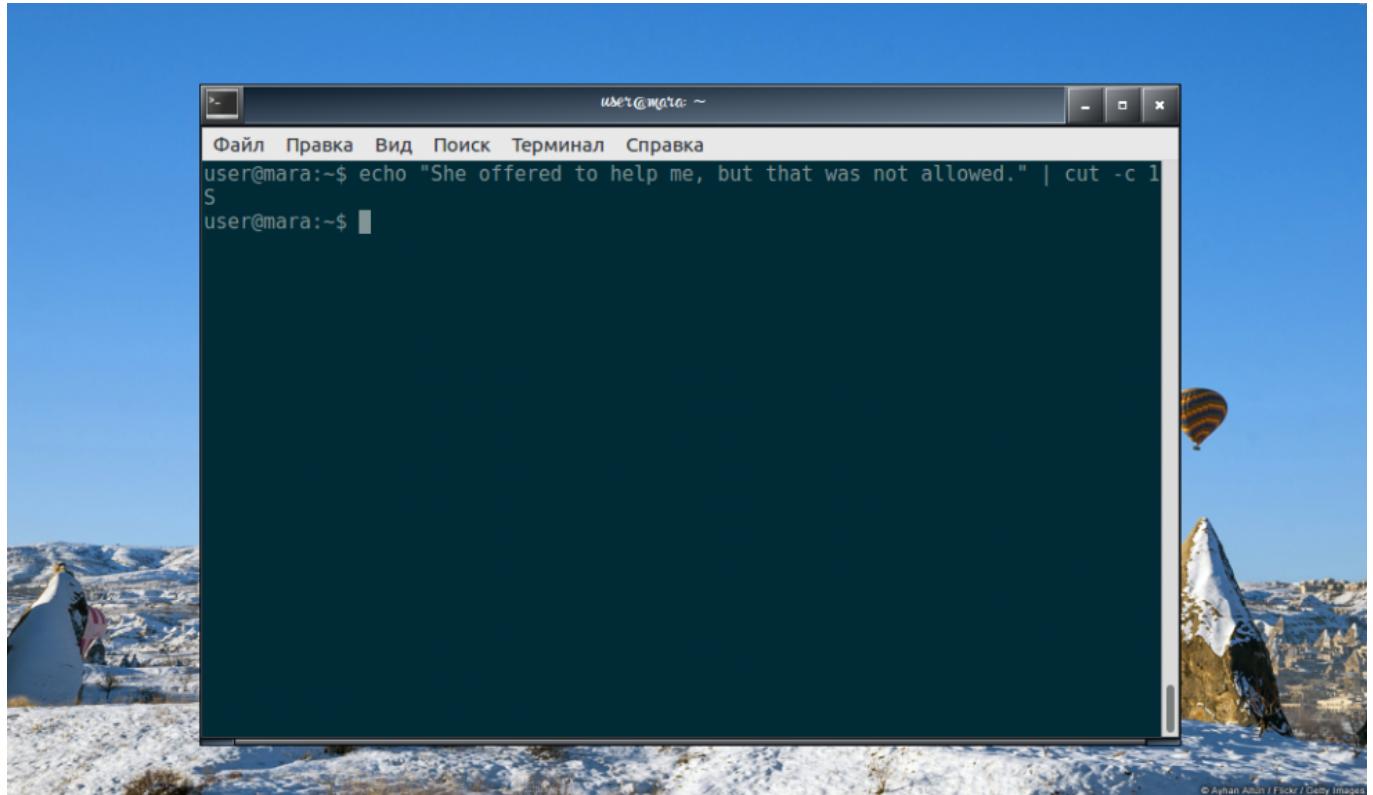
```
$ echo "The pallid sunlight through the window shone upon my hands." | cut -b 35-
```



2. Работа с символами

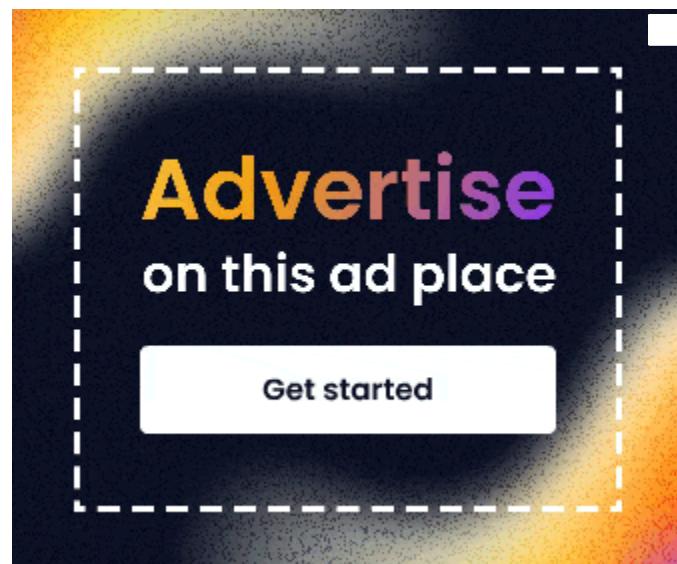
Тем, кто обрабатывает огромные массивы текстовых данных, часто приходится использовать команду `cut`, задавая необходимые параметры с помощью символов. Если текст напечатан через стандартный ввод, вырезать первый символ можно так:

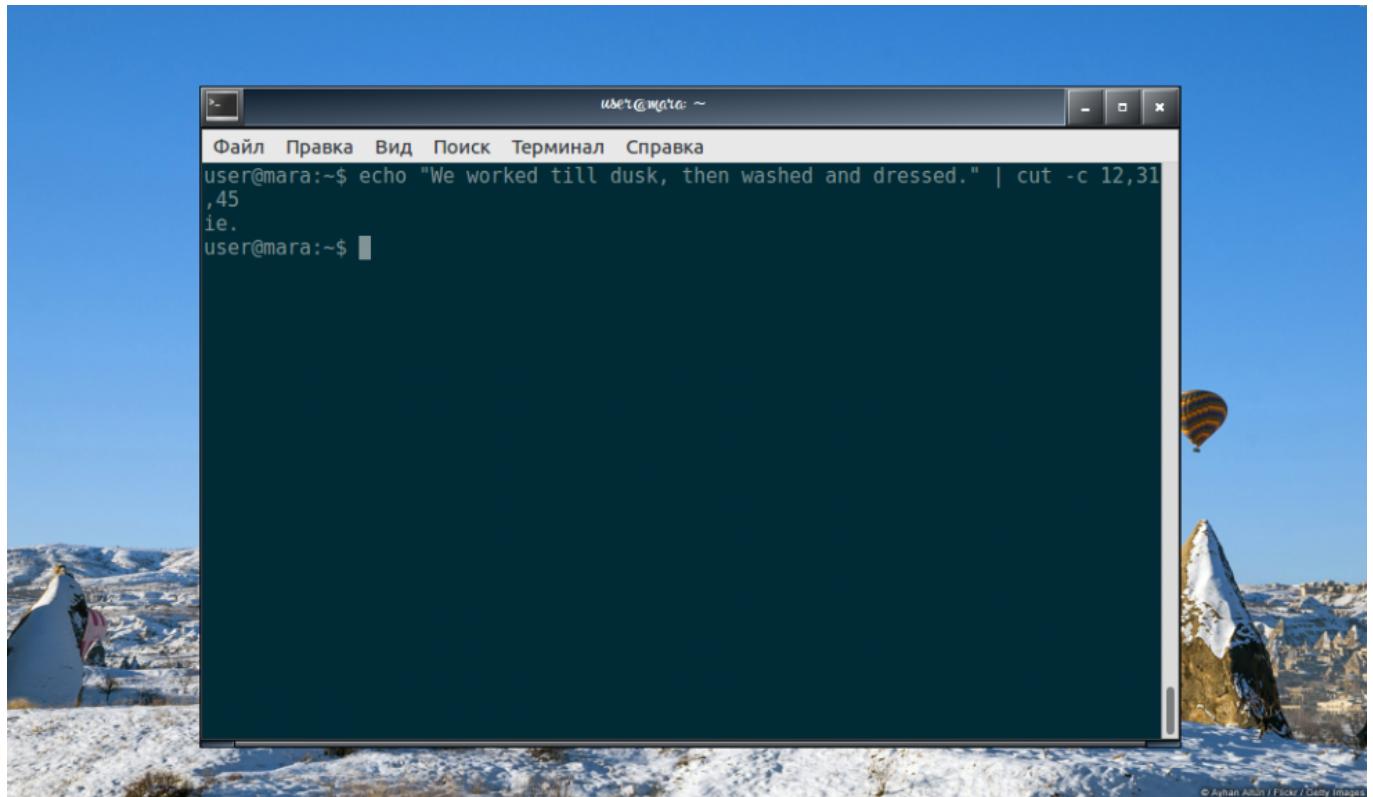
```
$ echo "She offered to help me, but that was not allowed." | cut -c 1
```



Чтобы вырезать другие символы, следует указать их порядковые номера. Важно помнить о том, что если числовое значение превысит количество символов в строке, команда не будет выполнена.

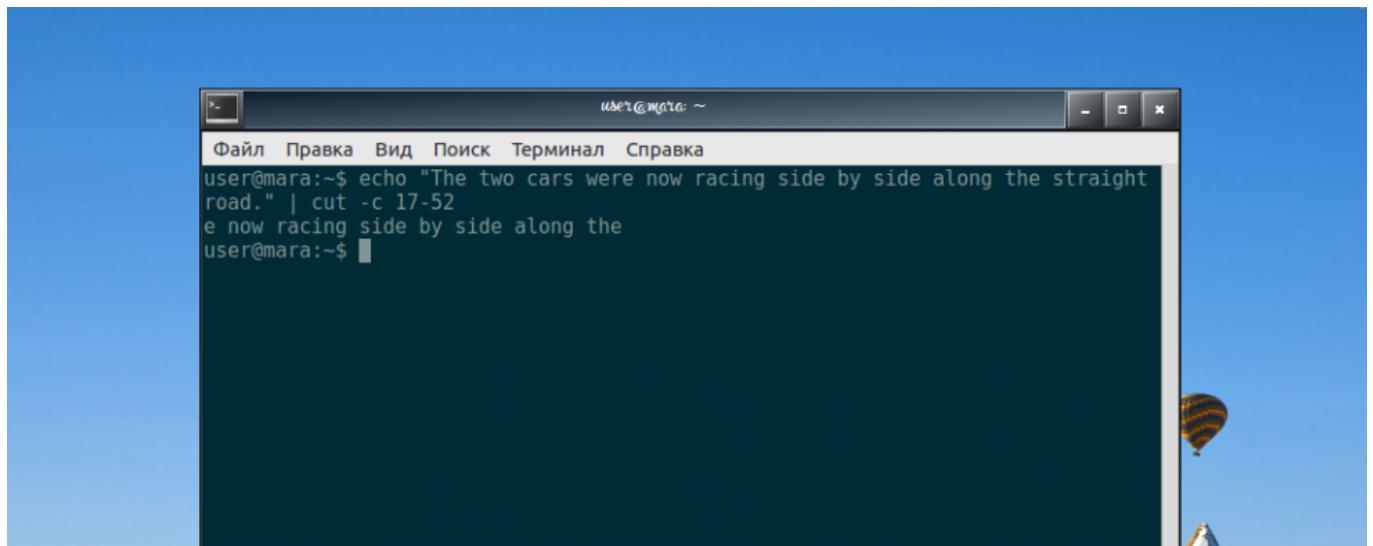
```
$ echo "We worked till dusk, then washed and dressed." | cut -c 12,31,45
```





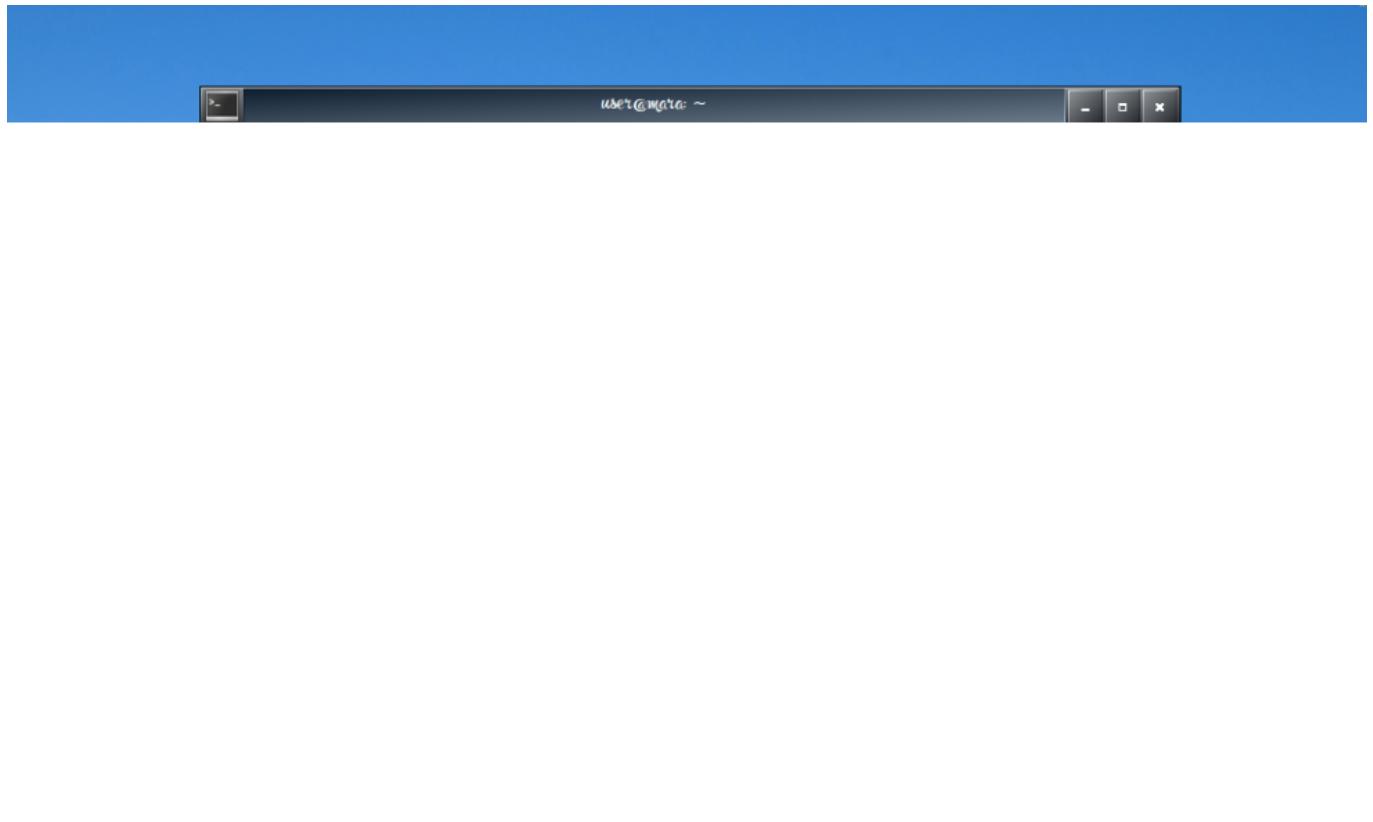
При работе с символами можно использовать диапазоны значений, точно так же, как и при работе с байтами.

```
$ echo "The two cars were now racing side by side along the straight road." | cut -c 17-52
```



Нередко возникает необходимость вырезать часть текста из готового документа. Чтобы сделать это, достаточно выполнить команду

```
$ cut -c 18,25 cut_command_example.txt
```



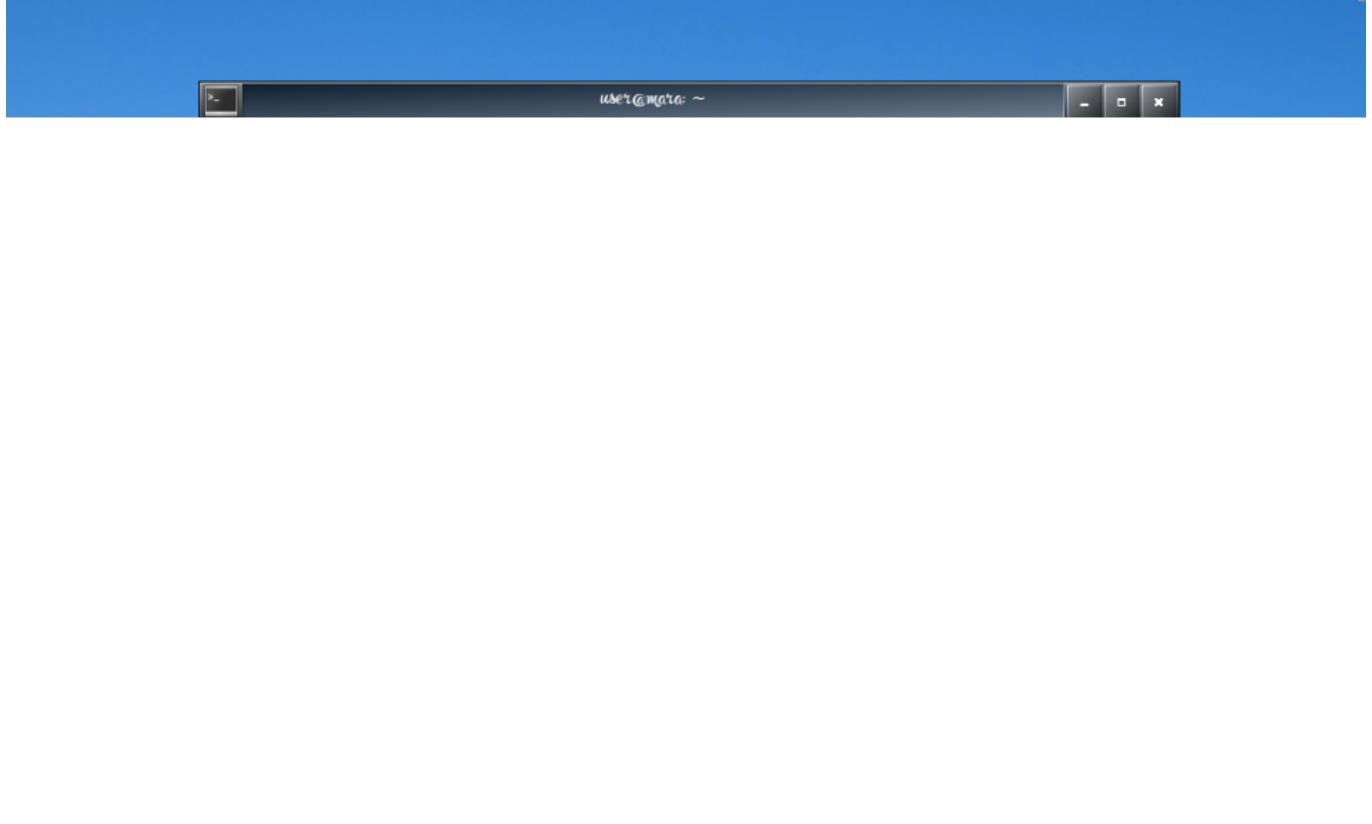
Как видно на скриншоте, были вырезаны восемнадцатый и двадцать пятый символы в каждой из строк.

Если же вырезать только первый символ в каждой строке при помощи команды

```
$ cut -c 1 cut_command_example.txt
```

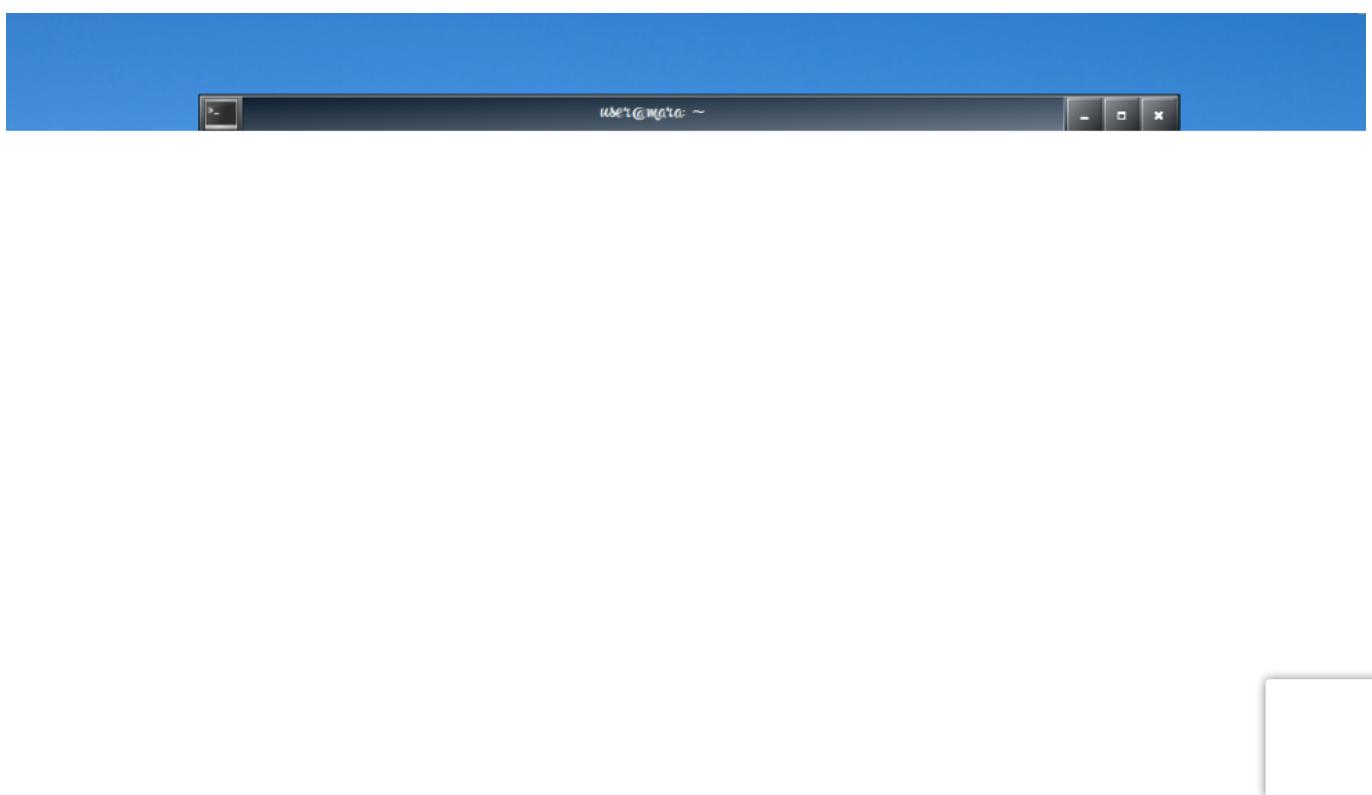
результат будет таким же, как и у команды

```
$ cat cut_command_example.txt | cut -c 1
```



Чтобы вырезать из текстового файла диапазон символов, нужно напечатать в терминале следующую команду:

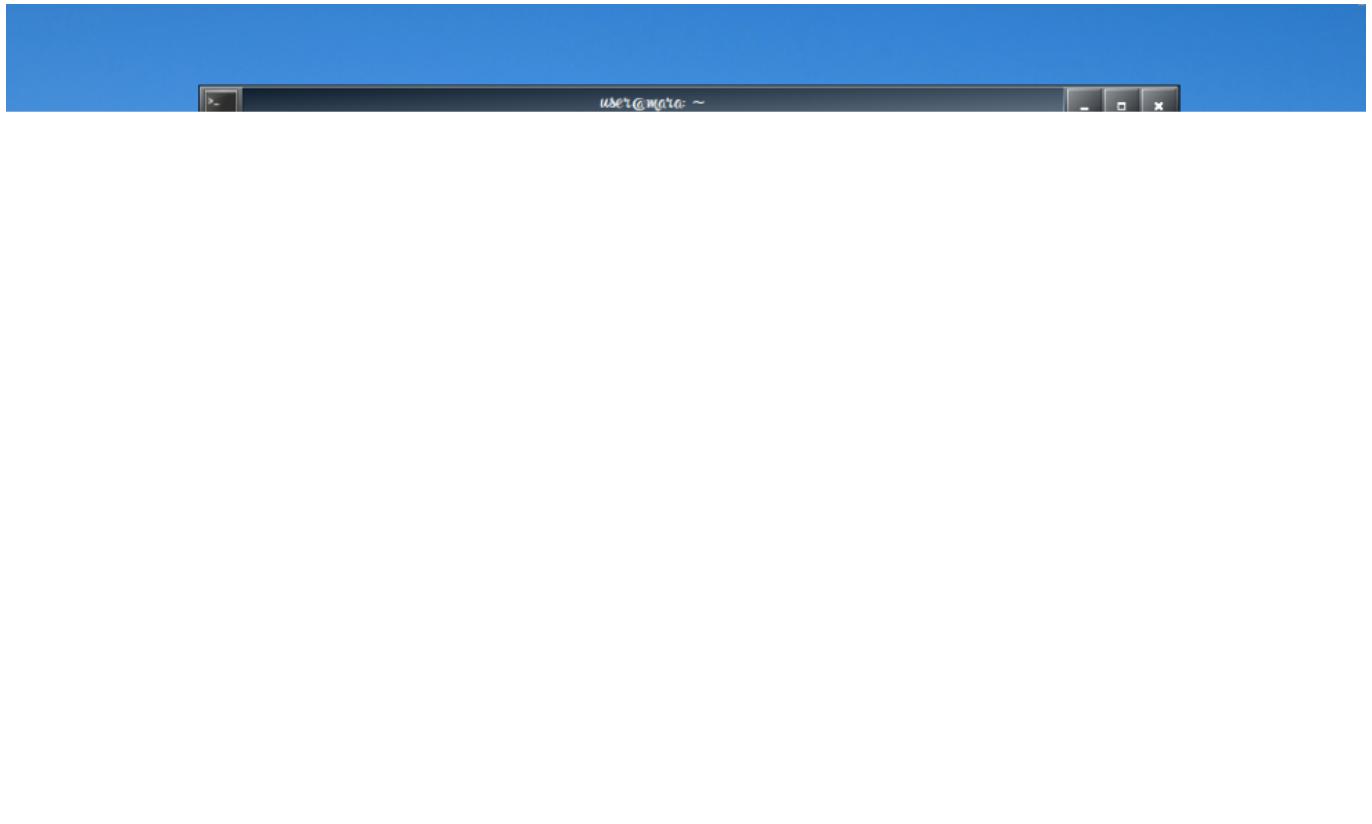
```
$ cut -c 7-34 cut_command_example.txt
```



Возможность сортировки вырезанных символов сохраняется независимо от того, в базе или в символах команде `cut` были заданы параметры текста. Команда `sort` рассортирует вырезанные символы.

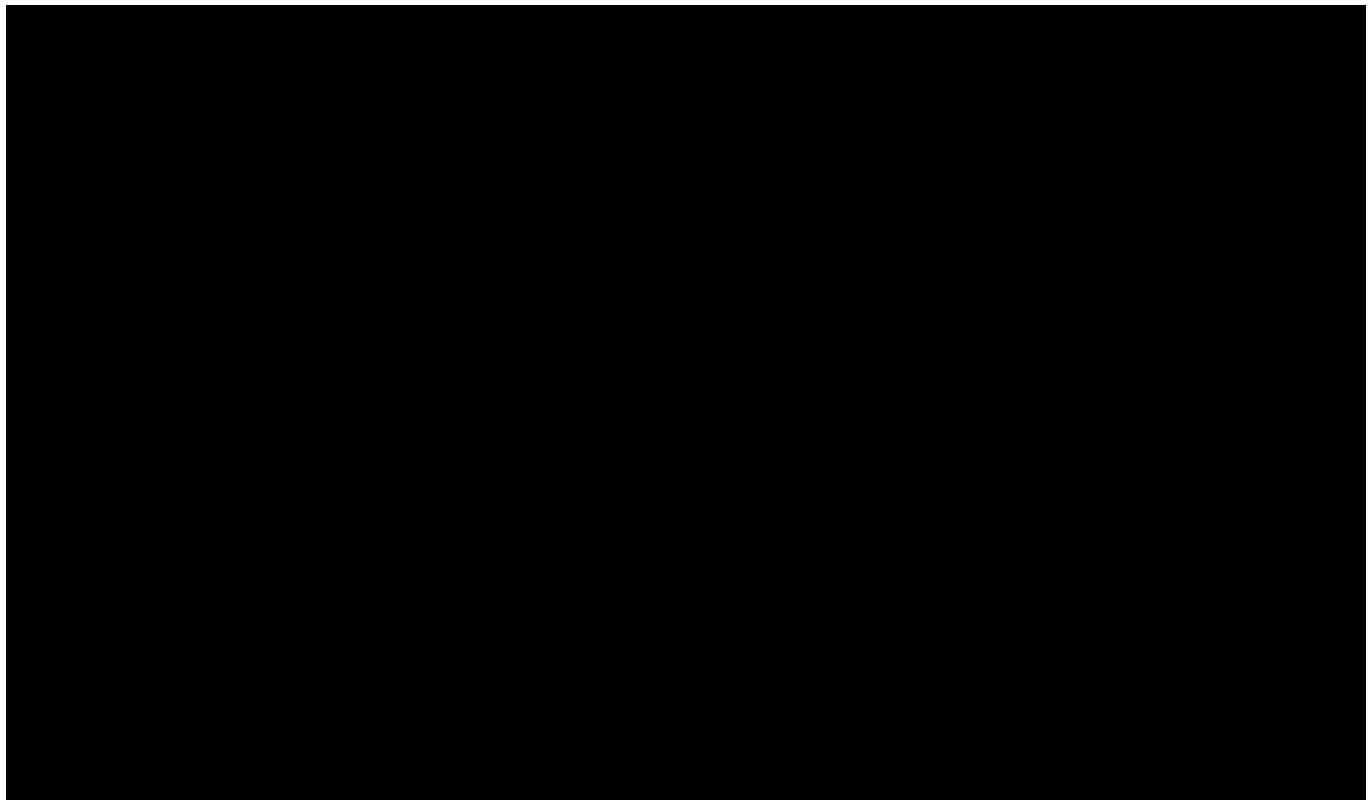
символы в алфавитном порядке по умолчанию:

```
$ cut -c 1-9 cut_command_example.txt | sort
```



Расстановка символов в обратном порядке – начиная с конца алфавита – требует применения опции -r:

```
$ cut -c 1-9 cut_command_example.txt | sort -r
```



Уже известный нам трюк с дефисом возле числа можно повторить и в данном случае:

```
$ echo "We had had fights for the same reason before now." | cut -c 12-
```

Будет вырезана последовательность символов, начиная с двенадцатого и заканчивая последним в строке.

А чтобы вырезать символы, начиная с первого и заканчивая двенадцатым, дефис следует переместить влево:

```
$ echo "We had had fights for the same reason before now." | cut -c -12
```

С текстом, содержащимся в файле, поступают аналогично:

```
$ cut -c -8 cut_command_example.txt
```

```
$ cut -c 8- cut_command_example.txt
```

Команда вырезает текст в каждой из строк, согласно заданным параметрам.

3. Работа со столбцами

Создавая файл `cut_command_example.txt`, мы отделяли слова друг от друга не только пробелами, но и двоеточиями. Наконец пришло время использовать эти знаки на практике. А нужны они для того, чтобы вырезать фрагменты текста из столбцов и называются разделителями. Кроме двоеточий и пробелов в качестве разделителей можно использовать дефисы.

Но начнем, как всегда, с текста в стандартном выводе. Чтобы вырезать слово из первого столбца, воспользуемся командой

```
$ echo "From the inn issued a smell of frying liver." | cut -d ' ' -f 1
```

В случае с текстовым файлом результат такой же – первое слово каждого столбца напечатано в терминале:

```
$ cut -d ':' -f 1 cut_command_example.txt
```

Чтобы вырезать несколько столбцов, нужно указать их номера. Для текста, напечатанного через стандартный ввод, команда имеет вид:

Privacy

```
$ echo "He was extraordinarily particular about politeness in others." | cut -d ':' -f 1,2,3
```

В данном примере ее результатом должны быть первые три слова – He was extraordinarily.

При использовании текстового файла аналогичная команда выглядит так:

```
$ cut -d ':' -f 1,2,3 cut_command_example.txt
```

Из файла были вырезаны три первые столбца, а их содержимое выведено в терминал.

Ничуть не хуже команда работает, если задать ей параметры текста в виде диапазона чисел:

```
$ echo "Surprised, we looked at one another." | cut -d ' ' -f 1-5
```



Как и ожидалось, были вырезаны столбцы (слова) с первого по пятый. Обратите внимание на два апострофа, которые из раза в раз появляются в записи команды при работе с текстом из стандартного ввода. Эти знаки необходимы, если разделителем является символ пробела. Теперь вырежем диапазон полей из файла:

```
$ cut -d ':' -f 1-3 cut_command_example.txt
```

Точно так же, как раньше сортировали вырезанные символы, мы можем поступить и сейчас – расставить названия сезонов из файла в алфавитном порядке:

```
$ cut -d ':' -f 1 cut_command_example.txt | awk '{print $ 1}' | sort
```

Меняя числовое значение опции `-f`, можно вырезать и сортировать содержимое других столбцов. Отсортировать названия сезонов в обратном порядке поможет опция `-r`, добавленная к команде `sort`:

```
$ cut -d ':' -f 1 cut_command_example.txt | awk '{print $ 1}' | sort -r
```



Вырезать содержимое, начиная с определенного столбца и заканчивая последним в строке, получится, если использовать такую команду, где справа от числа, обозначающего номер начального столбца, стоит дефис:

```
$ echo "We became still more friendly." | cut -d ' ' -f 2-
```

Переместив дефис влево, мы сделаем этот столбец последним, а вырезать слова команда будет, начиная с первого столбца.

```
$ echo "We became still more friendly." | cut -d ' ' -f -2
```

Работа этой команды с текстовыми файлами не имеет принципиальных отличий:

```
$ cut -d ':' -f 2- cut_command_example.txt
```

```
$ cut -d ':' -f -2 cut_command_example.txt
```

Команда `cut` также пригодится при работе с массивными CSV-документами. Ее синтаксис и опции в этом случае остаются неизменными. Например, при таком написании будут вырезаны первые два столбца файла с названием `file_example.csv`:

```
$ cut -d ',' -f 1,2 file_example.csv
```

Выходы

Команда `cut linux` – это инструмент, который пригодится всем пользователям Linux, которые работают с большими текстовыми документами. Ее возможности поистине фантастические и она имеет немало преимуществ перед другими схожими утилитами.

Как всегда, не стесняйтесь задавать вопросы в комментариях, если какие-то аспекты работы этой команды остались непонятными для вас.

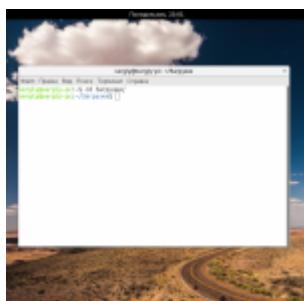
Была ли эта информация полезной для вас?

Да

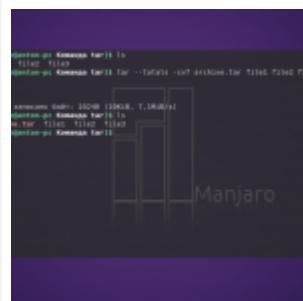
Нет

X

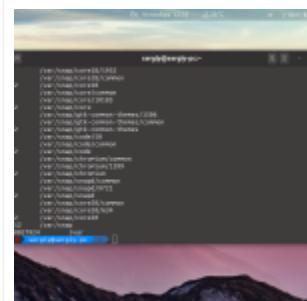
Похожие записи



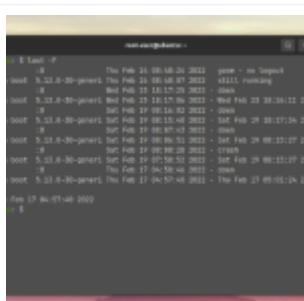
Команда `cd` в Linux



Команда `tar` в Linux



Команда `du` в Linux



Команда `last` в Linux

Оцените статью

Privacy

 (13 оценок, среднее: 3,92 из 5) [Команды](#)

Об авторе

**Автор:** Rubaiat Hossain**Источник:** <https://www.ubuntupit.com/simple-and-useful-linux-cut-command-in-unix/>**Переводчик:** [ellado](#)

Больше восьми лет назад мною было принято решение объявить бойкот оконной монополии и установить на свой компьютер Ubuntu. С тех пор это моя основная ОС. Иногда в порядке эксперимента "подселяю" к ней собратьев из семьи Linux. Увлекаюсь фотографией и горным туризмом. В свободное от работы время пишу статьи для losst.ru.

10 комментариев к “Команда cut Linux”

**Михаил**[22 декабря, 2019 в 9:57 пп](#)

Когда в ГУЕ прог тьма, то она на фиг не нужна, но знать её надо.?

[Ответить](#)**Энтони**[23 декабря, 2019 в 12:47 пп](#)

Да. Такое впечатление, будто вы не представляете работу в GNU без графической оболочки и где вообще можно применять консольные утилиты.

[Privacy](#)

[Ответить](#)**Павел**26 декабря, 2019 в 6:52 пп

Это надо для написания скриптов в основном

[Ответить](#)**DIGITAL**24 августа, 2020 в 8:50 пп

товарищи, а как вырезать предпоследнее и последнее поле, если не знаешь общее количество полей?

(например, uslugi.mos.ru -> mos.ru или т.рассылка.vk.com ->vk.com)

[Ответить](#)**DIGITAL**24 августа, 2020 в 8:58 пп

нашел,

```
awk -F. '{print $(NF-1) "." $NF}'
```

но столкнулся с проблемой если запись с номером (FN-1) отсутствует, выдаёт ошибку

[Ответить](#)**emriu**11 декабря, 2020 в 3:33 пп

Через rev:

```
echo " One Two Three" | rev | cut -d ' ' -f -2 | rev
```

[Privacy](#)

```
> Two Three
echo " One Two Three Four Five" | rev | cut -d ' ' -f -2 | rev
> Four Five
```

[Ответить](#)



Андрей

2 сентября, 2022 в 2:21пп

Формирую текстовую переменную:

```
OUTPUT ="$(ls -l /dev/ | grep sd)"
```

Получаю:

```
echo "${OUTPUT}"
```

```
brw-rw---- 1 root floppy 8, 32 сент 2 06:28 sdc
```

```
brw-rw---- 1 root floppy 8, 33 сент 2 06:28 sdc1
```

Задача - присвоить переменной пусть той же OUTPUT значение sdc1 в конце текста...

Там может быть sda1 и sdd1...

Хочется далее выделенные символы использовать в команде:

```
sudo mount /dev/sdc1 /mnt
```

Подскажите как это сделать, кто знает!?

[Ответить](#)



Андрей

2 сентября, 2022 в 2:22пп

Ну или решить данную проблему другим способом как???

[Ответить](#)



Андрей(другой)

2 октября, 2022 в 4:34пп

Про опции -с и -п не написали.

[Ответить](#)



Андрей(другой)

2 октября, 2022 в 4:36 pp

Ошибся, не "-с", а "-s".

[Ответить](#)

Оставьте комментарий

Имя *

Email

Я прочитал и принимаю политику конфиденциальности. Подробнее [Политика конфиденциальности](#) *

Комментировать

Privacy

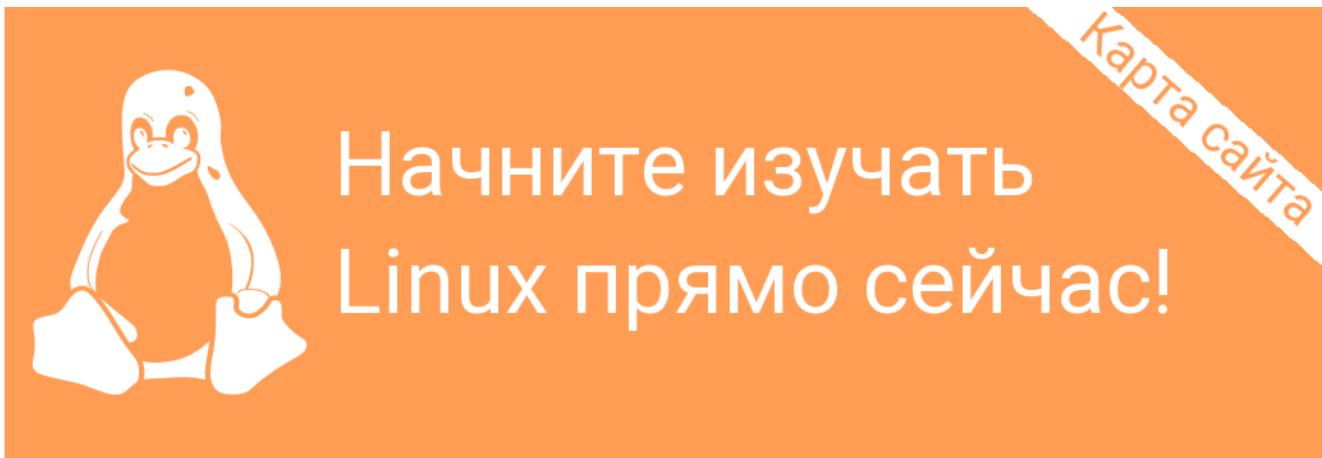
Русский

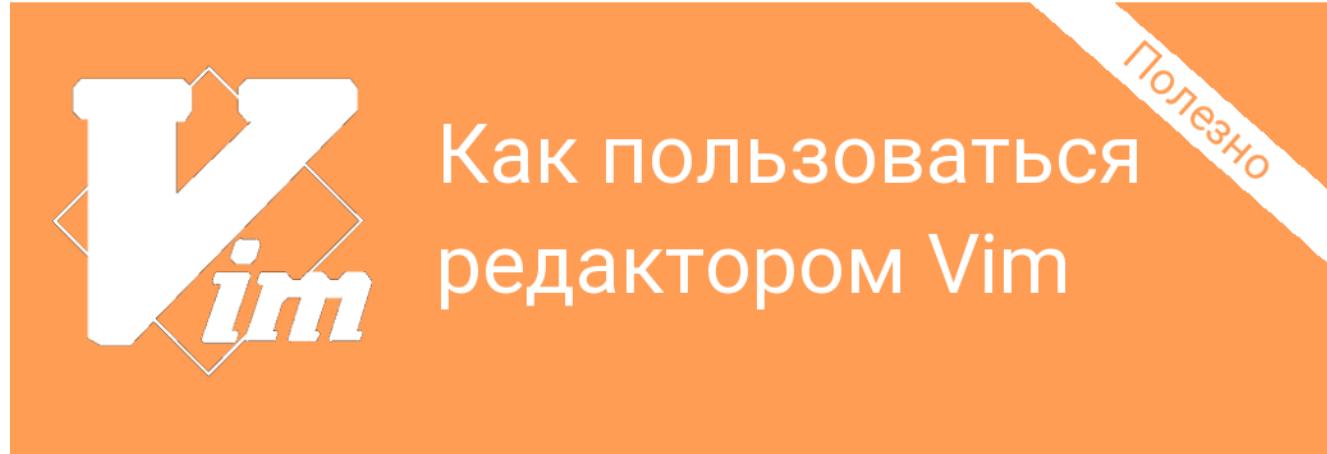
Поиск

ПОИСК ПО КОМАНДАМ

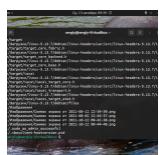
Начните вводить команду

Поиск



[Лучшие](#)[Свежие](#)[Теги](#)[Команда chmod Linux](#)

2020-04-13

[Команда find в Linux](#)

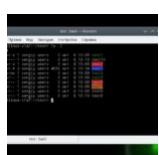
2021-10-17

[Как узнать IP-адрес Linux](#)

2023-04-14

[Настройка Сроп](#)

2021-10-01

[Права доступа к файлам в Linux](#)

2020-10-09

[Privacy](#)

РАССЫЛКА

Ваш E-Mail адрес

Я прочитал(а) и принимаю политику конфиденциальности

Sign up

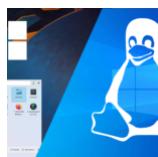
Windows

Списки



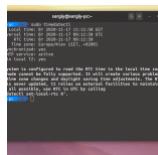
Восстановление Grub после установки Windows 10

2020-08-15



Установка Linux рядом с Windows 10 или 11

2023-02-08

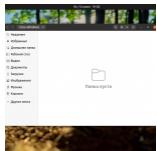


Сбивается время в Ubuntu и Windows

2023-02-18

Ошибка Ubuntu не видит сеть Windows

Privacy



2023-02-18

[Смотреть ещё](#)

МЕТА

[Регистрация](#)[Войти](#)[Лента записей](#)[Лента комментариев](#)

СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ



Интересное



6 причин, почему Ubuntu лучше Windows

2021-10-01

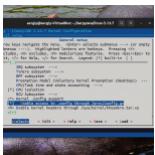


Privacy



Лучшие темы курсоров Linux

2020-12-18



Сборка ядра Linux

2021-08-14



Использование Docker для чайников

2021-04-08

©Losst 2024 CC-BY-SA [Политика конфиденциальности](#)



Privacy