



[Главная](#) >> [Команды](#) >> Команда date в Linux

Команда date в Linux

Обновлено: 13 апреля 2020 **Опубликовано:** 22 мая, 2018 от [Geradot](#) , 4 комменариев,
время чтения: 4 минут

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

Главное свойство утилит GNU/Linux - делать что-то одно, но эффективно. Яркий пример – команда **date** Linux, работающая с датой и временем. С её помощью можно извлекать любую дату в разнообразном формате, в том числе и рассчитывать прошлое и будущее время. Привилегированные пользователи могут перезаписывать системное время, используя её.

Утилита предустановлена во всех дистрибутивах GNU/Linux. В этой статье будут рассмотрены возможности **date** и способы применения этой команды.

Содержание статьи

- [Синтаксис команды date](#)
- [Примеры использования date](#)
- [Выводы](#)

Синтаксис команды date



Программа может выполняться от имени обычного пользователя. Стандартный синтаксис команды (квадратные скобки обозначают необязательное наличие):

```
date [ОПЦИИ] ... [+ФОРМАТ]
```

Ниже представлена таблица с часто применяемыми опциями для **date**.

Опция	Длинный вариант	Значение
-d STRING	--date=STRING	Вывод даты по указанной строке (например 'yesterday', 'tomorrow', 'last monday').
-I	--iso-8601[=FMT]	Вывод даты в формате ISO 8601. FMT по умолчанию содержит 'date'. Также может содержать 'hourse', 'minutes', 'seconds', 'ns' для отображения соответствующих значений и часовой пояс относительно UTC рядом с датой.
	--rfc-3339=FMT	Вывод даты в формате RFC 3339. FMT по умолчанию содержит 'date'. Также может содержать 'seconds' и 'ns' для отображения секунд или наносекунд.
-r FILE	--reference=FILE	Вывод даты последней модификации указанного файла в формате по умолчанию.
-u	--utc	Вывод UTC-даты

Аргумент **ФОРМАТ** отвечает за форматирование вывода даты. Для его указания необходимо поставить знак "+" и написать нужную маску. Наиболее популярные форматы:

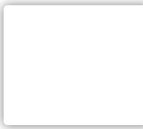
Формат	Значение
%%	Знак процента
%a	День недели текущей локали в короткой форме ("Чтв")
%A	День недели текущей локали в длинной форме ("Четверг")
%b	Месяц года текущей локали в короткой форме в родительном падеже ("январ")
%B	Месяц года текущей локали в длинной форме в родительном падеже ("января")
%c	Дата и время текущей локали без указания часового пояса
%C	Первые две цифры текущего года
%d	Числовой день месяца с ведущим нулём

Формат	Значение
%D	Дата в формате %m/%d/%y
%e	День месяца; аналог %_d
%F	Дата в формате %Y-%m-%d
%h	Аналог %b
%H	Часы (00..23)
%I	Часы (01..12)
%j	День года (001..366)
%m	Месяц (01..12)
%M	Минуты (00..59)
%n	Новая строка
%q	Квартал года
%S	Секунды (00..59)
%t	Знак табуляции
%T	Время в формате %H:%M:%S
%u	Числовой день недели; 1 – понедельник
%x	Дата в локальном формате
%X	Время в локальном формате
%Z	Аббревиатура временной зоны

Примеры использования date

Введем команду без параметров.

```
$ date
```



Будет отображена текущая дата и время в соответствии с настройками локали системы



```
[anton@anton-pc ~]$ date
Чтв мая 17 12:14:32 EEST 2018
[anton@anton-pc ~]$
```



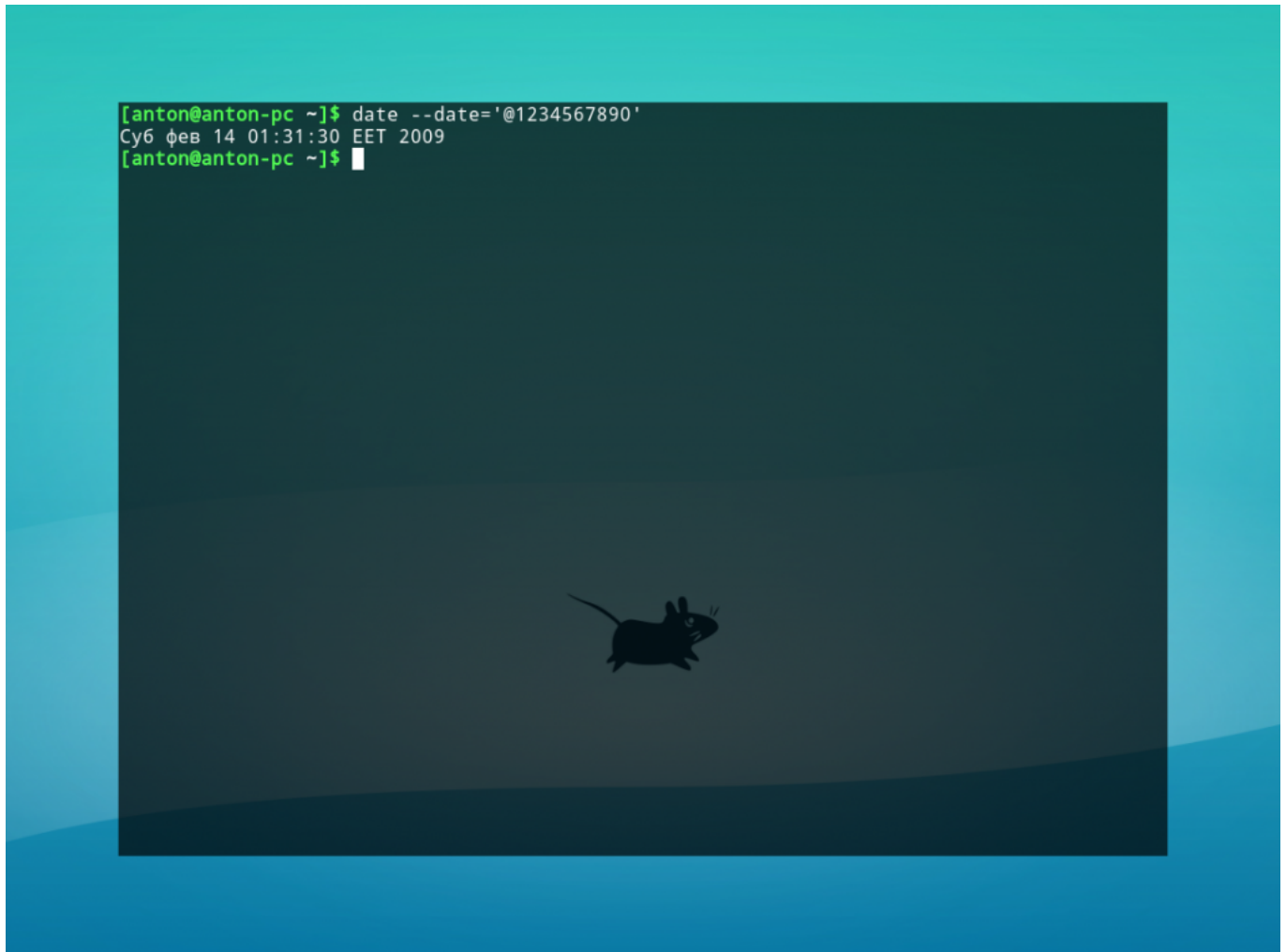
Команда **date** без параметров по умолчанию применяет маску **%a %b %d %X %Z**. Поскольку все форматы должны быть переданы как один параметр (из-за принципа обработки данных командным интерпретатором Bash), пробелы между ними необходимо экранировать обратным слэшем (\) или взять в кавычки.

Особое внимание следует уделить параметру **-d (--date)**. Его функциональность не слишком очевидна, но при этом наиболее обширна.

Пример 1. Вычисление даты по числу секунд, прошедших с 1 января 1970 года.

Privacy

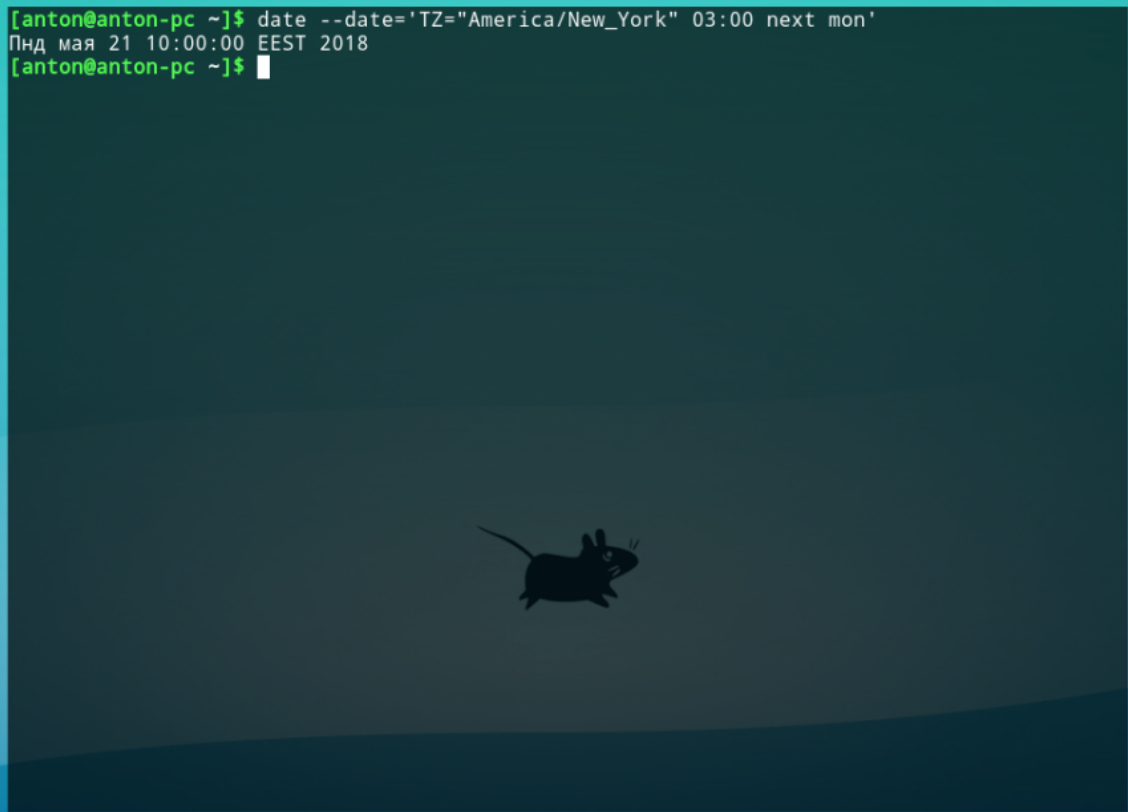
```
$ date --date='@1234567890'
```



Пример 2. Вычисление даты и времени следующего понедельника при указании часового пояса Нью-Йорка в 03:00.

```
$ date --date='TZ="America/New_York" 03:00 next mon'
```





```
[anton@anton-pc ~]$ date --date='TZ="America/New_York" 03:00 next mon'
Пнд мая 21 10:00:00 EEST 2018
[anton@anton-pc ~]$
```

Обратите внимание: указывать название дня недели или месяца можно в любом регистре, в короткой или длинной форме. Параметры **next** и **last** обозначают следующий и прошедший, соответственно, ближайшие дни недели.

Пример 3. Если текущий день месяца – последний, сформировать отчет о занятости дискового пространства корневого и домашнего каталога в файл **report**.

```
$ #!/bin/bash
if [[ $(date --date='next day' +%d) = '01' ]]; then
df -h / /home > report
```

Такой скрипт можно использовать для автоматизации работы с помощью демона **crontab** или **anacron**.

Выводы

Команда **date** Linux является эффективным инструментом работы с датой и временем, с широкой возможностью их расчёта для прошедших или будущих показателей. Также он применяется в написании сценариев в командном интерпретаторе Bash.

Была ли эта информация полезной для вас?

Да

Нет

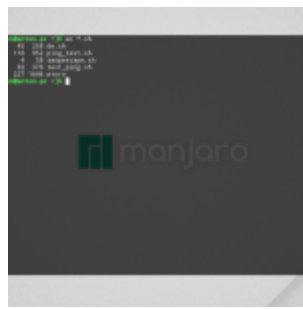
X

Похожие записи





Команда ls Linux



Команда wc в Linux



Команда tr в Linux



Команда tail Linux

Оцените статью

★★★★☆ (7 оценок, среднее: **3,00** из 5)



Статья распространяется под лицензией Creative Commons ShareAlike 4.0 при копировании материала ссылка на источник обязательна .

📁 [Команды](#)

🏷️ [команды](#)

Об авторе



GERADOT

Меня зовут Антон. Сажу на GNU/Linux с 2011 года. На данный момент остановился на Manjaro.



Privacy

4 комментария к “Команда date в Linux”

**Dou**22 мая, 2018 в 9:25 пп

Автор Антон, 7 лет на linux, я обещал Adminu молчать, молчу....

[Ответить](#)**Серёга**11 ноября, 2018 в 10:05 пп

Да, не, удобно, когда все команды описаны с примерами и по отдельности. Удобно начинающим; я раньше кучу ссылок держал на подобные статьи, сейчас уже и так запомнилось. В ман'ах не всегда на русском и не всегда с толковыми примерами...

Так что, автор - продолжайте в том же духе, команд/утилит ещё много, смотрю - уже начинает набираться количество ваших статей. Оценить не могу - регистрироваться лень...

[Ответить](#)**Славик**18 апреля, 2021 в 2:11 пп

а как вывести только время без даты

[Ответить](#)

**Starik**28 мая, 2021 в 11:36 пп

В терминале:

```
date +%T
```

В скрипте:

```
#!/bin/sh
```

```
CURRENTTIME=$(date +%T)
```

```
echo $CURRENTTIME
```

или:

```
#!/bin/sh
```

```
echo $(date +%T)
```

[Ответить](#)

Оставьте комментарий

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

Комментировать

Privacy


Русский

Поиск

ПОИСК ПО КОМАНДАМ

Начните вводить команду

Поиск



Начните изучать
Linux прямо сейчас!

Карта сайта



Privacy



Как пользоваться редактором Vim

Полезно

Лучшие

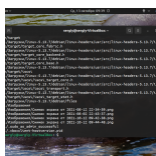
Свежие

Теги



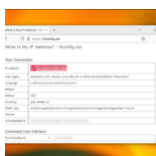
Команда chmod Linux

2020-04-13



Команда find в Linux

2021-10-17



Как узнать IP-адрес Linux

2023-04-14



Настройка Cron

2021-10-01



Права доступа к файлам в Linux

2020-10-09



Privacy

РАССЫЛКА

Ваш E-Mail адрес

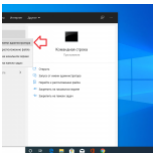
☐ Я прочитал(а) и принимаю политику конфиденциальности

Sign up



Windows

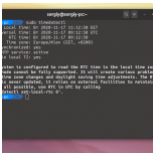
Списки



Восстановление Grub после установки Windows 10
2020-08-15



Установка Linux рядом с Windows 10 или 11
2023-02-08

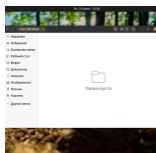


Сбивается время в Ubuntu и Windows
2023-02-18

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

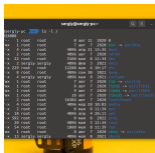


Интересное

[Чем отличается Chrome от Chromium](#)

2021-11-23

[Privacy](#)



Как правильно: папка или каталог в Linux

2022-05-10



Команды Linux для работы с файлами

2020-04-14



5 причин использовать Firefox

2021-12-09

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

