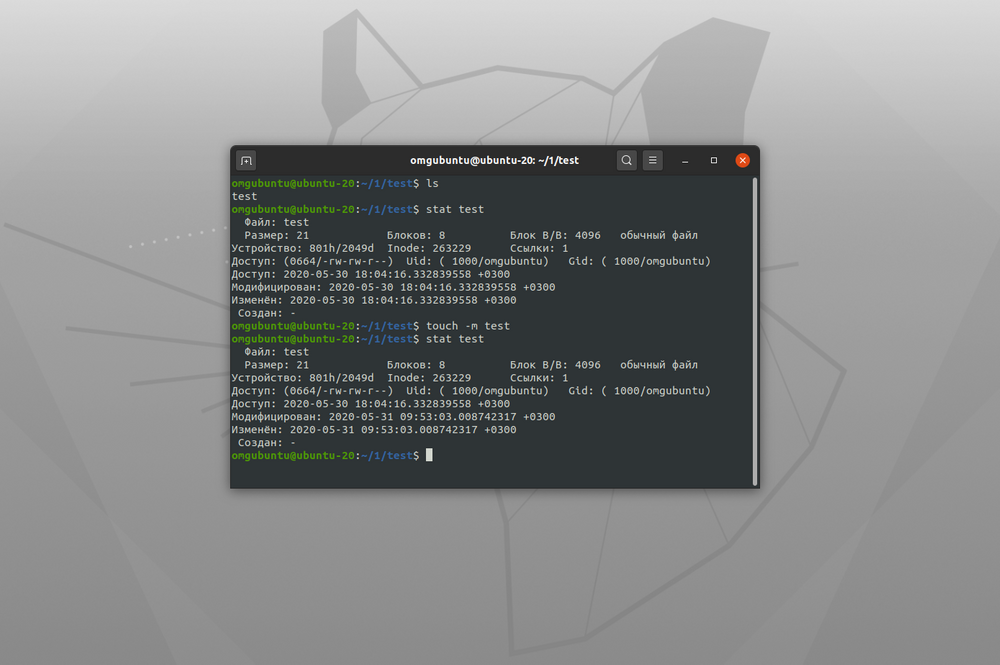
**Источник: https://omgubuntu.ru/15-polieznykh-primierov-ispolzovaniia-komandy-touch-v-sistiemakh-linux/**

**15 полезных примеров использования команды touch в Linux системах**

By **[Andrew E.](https://omgubuntu.ru/author/andrew/)** Июнь 05, 2020 6 мин чтения



Команда touch - это простой, но полезный инструмент в Unix-подобных системах, который позволяет пользователям создавать файлы и изменять временные метки непосредственно из терминала. Это одна из самых фундаментальных терминальных команд в Linux, и сисадмины часто используют ее для различных целей. Перед вами краткое руководство, содержащее 15 практических примеров ее использования.

**Практическое использование команды touch для пользователей Linux**

Приведенные ниже примеры помогут вам получить полное представление о команде. Однако, вы должны попробовать использовать эти команды сами, если хотите освоить этот инструмент в целом. Итак, запустите ваш терминал Linux и начните выполнять команды, приведенные ниже, по мере того, как вы о них будете читать.

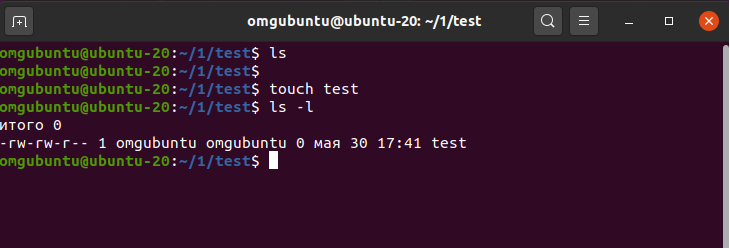
**1. Создание пустого файла**

При самом простом использовании touch создает простой файл, который ничего не содержит. Он очень полезен в ряде ситуаций, например, когда вы экспериментируете с файловыми операциями или какими-либо другими командами терминала. Для этого просто передайте имя файла после команды touch.

$ touch test

Так в текущей рабочей директории создается простой пустой файл, который называется test. Вы можете проверить это с помощью команды ls. Вы также можете использовать команду touch для создания пустого файла в любой другой директории, просто передав правильный путь перед именем файла.

$ touch /путь/к/файлу



**2. Создание нескольких пустых файлов**

С помощью данной команды можно также создавать несколько файлов одновременно из одного терминала. Все, что вам нужно сделать, это передать все имена файлов, которые вы хотите создать один за другим. Взгляните на приведенную ниже команду, чтобы увидеть, как это работает.

$ rm test

$ touch test1 test2 test3 test4 test5

Вышеуказанная команда создаст все пять файлов одновременно. Проверить это можно с помощью команды **ls -l** в терминале. Вы также можете использовать фигурные скобки для создания нескольких файлов одновременно с помощью touch, как показано в примере, приведенном ниже.

$ rm test1 test2 test3 test4 test5 # сначала удалите файлы

$ touch test{1,2,3,4,5}

Вышеуказанная команда сделает ту же работу, но гораздо более аккуратным способом.

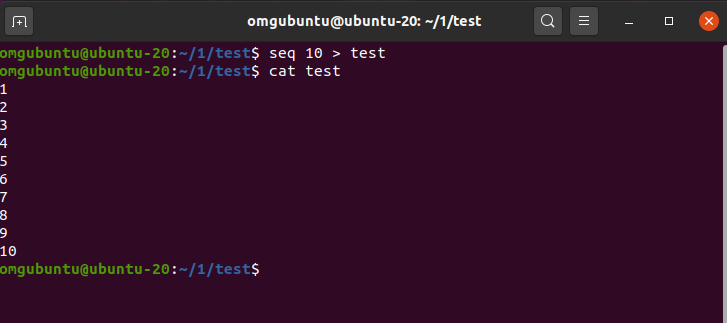
**3. Создание и заполнение файла**

В приведенном ниже примере показано, как создать простой файл с помощью сенсорного ввода и заполнить его некоторым содержимым. Для этой цели мы используем команду **seq** в Linux. Но вы можете прибегнуть и к другим методам, если хотите.

$ rm test1 test2 test3 test4 test5         # сначала удалите файлы

$ seq 100 > test

Теперь тестовый файл не пустой, а содержит числа от 1 до 10. Вы можете проверить это с помощью команды Linux cat.



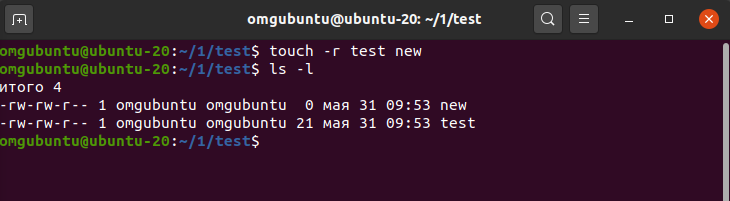
**4. Повторение метки времени последнего доступа к файлу**

Команда touch позволяет пользователям повторять метки времени времени последнего доступа разных файлов. Для этого необходимо использовать ключ -r. Мы создадим новый файл под названием new, и вместо использования текущего времени мы возьмем значения времени из тестового файла, созданного ранее.

$ touch -r test new

$ ls -l

Вы можете проверить это с помощью команды ls. Вы должны увидеть, что файл *new*имеет временную метку, более раннюю, чем текущее время, и такую же, как и файл *test*. Для этого ключа также существует длинная форма, называемая **-reference**.



**5. Изменение метки времени последнего доступа к файлу**

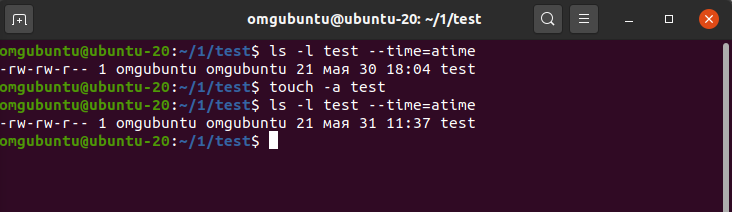
Администраторы Linux часто используют команду для изменения метки времени последнего доступа к файлам, созданным ранее. Следующая команда показывает нам, как это сделать. Для этого нам нужно использовать ключ **-a**. Сначала проверьте метку времени файла *test*, используя нижеприведенную команду.

$ ls -l test --time=atime

Теперь воспользуйтесь ключом **-a**, чтобы изменить эту метку времени с помощью команды touch. Обратите внимание, что ключ **-time=atime** из **ls** показывает нам время доступа к файлу.

$ touch -a test

Теперь, если вы еще раз проверите временную метку файла *test*, вы заметите, как она изменилась после выполнения вышеуказанной команды. Это один из наиболее распространенных видов использования в Linux.



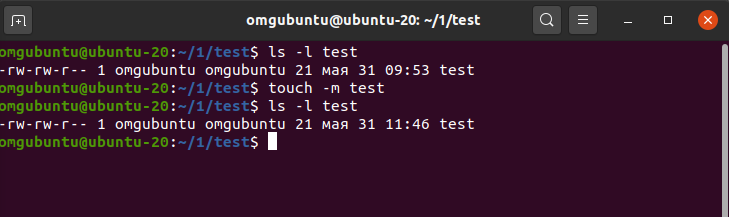
**6. Изменение метки времени последнего изменения файла**

Вы также можете изменить время последнего изменения файла в Linux с помощью этой команды. Для этого необходимо использовать ключ **-m**. Внимательно посмотрите на пример ниже, чтобы увидеть, как это работает.

$ ls -l test

$ touch -m test

Во-первых, мы просмотрели последнее время последнего изменения файла *test* с использованием команды **ls**. Затем мы использовали ключ **-m**вместе с командой touch, чтобы изменить это значение на текущее время. Проверить это можно, выполнив еще раз команду **ls**.



**7. Изменение даты и времени на текущее время**

Вы можете использовать ключи **-a** и **-m** вместе для изменения как метки времени последнего доступа к файлу, так и времени последнего изменения файла в Linux. Для демонстрации этого необходимо использовать старый файл. Вы можете просто запустить приведенную ниже команду для одного такого файла, чтобы посмотреть, работает ли это так, как ожидалось или нет.

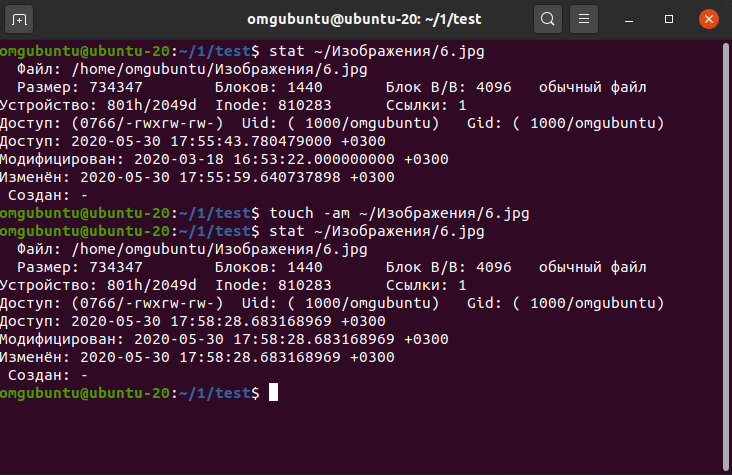
$ stat /путь/к/старому/файлу

Вы можете просмотреть всю информацию о времени, относящуюся к старому файлу, с помощью команды **stat**. Она покажет время последнего доступа, время последнего изменения и время последнего изменения атрибутов вашего файла. Теперь используйте следующую команду для изменения даты и времени на текущие.

$ touch -am /путь/к/старому/файлу

Проверьте изменение даты и времени еще раз с помощью команды **stat**.

$ stat /путь/к/старому/файлу



**8. Пропустить создание файла**

Ключ **-c** или **-no-create** позволяет пропустить создание файла. Хотя вы можете считать это непродуктивным, но многие пользователи используют этот ключ для проверки существования файла.

$ touch -c new-file

$ touch --no-create new-file

При выполнении вышеописанных команд в Вашей рабочей директории не произойдет никаких изменений. Вы можете запустить команду **ls** еще раз, чтобы проверить это.

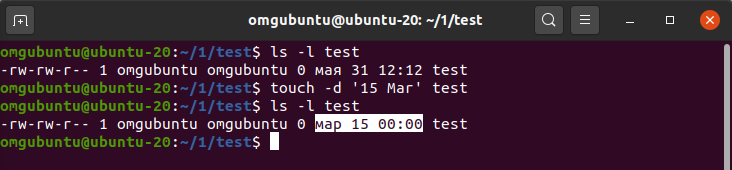
**9. Изменение даты файла**

Мы также можем использовать команду touch, чтобы указать дату файла вручную. Для этого необходимо использовать ключ -d или -date. Приведенная ниже команда показывает, как изменить дату файла на более раннюю с помощью touch.

$ ls -l test

$ touch -d '15 Mar' test

Теперь, если вы снова запустите команду ls, вы увидите, что она показывает дату файла - 15 Марта. Но вы же создали этот файл сегодня, верно? Если Вы посмотрите внимательно, то увидите, что теперь ls показывает только дату, а не время. Вы даже можете использовать эту команду, чтобы установить дату в будущем.



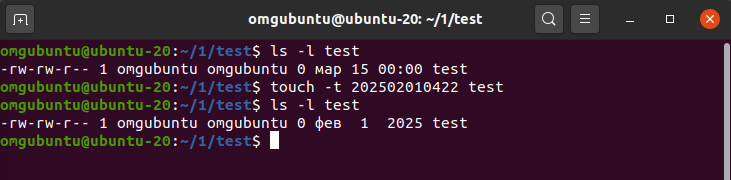
**10. Измение даты и времени на пользовательские значения**

Ключ **-t** позволяет нам изменять как дату, так и время файла на персонализированные значения. Таким образом, мы можем изменить несколько параметров, включая год, месяц, дату, час, минуты и секунды. Внимательно посмотрите на следующий пример, чтобы понять, как это работает.

$ ls -l test # покажет время и дату файла

$ touch -t 2025 02 01 04 22 test

Теперь снова запустите команду **ls**, чтобы проверить параметры даты и времени. Аргумент к ключу **-t** должен быть в формате YYMMDDhhmm. Так как мы установили значения, которые будут в будущем, команда ls не сможет показать параметр времени.



**11. Изменение даты с помощью слова**

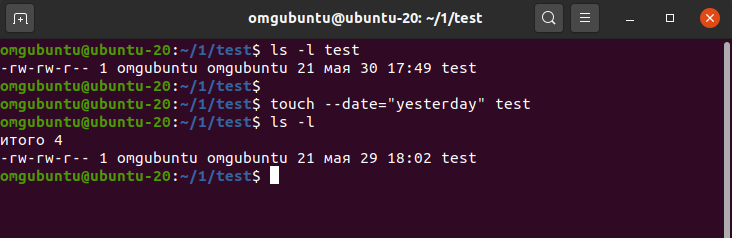
Также можно изменять дату файла в формате слов, например, "yesterday" с помощью команды touch. Для этого просто передайте аргумент ввиде строки к ключу **-date**. Команда ниже показывает, как изменить дату файла на предыдущий день.

$ touch --date="yesterday" test

Используйте следующую команду для изменения даты на следующий день.

$ touch --date="tomorrow" test

Проверьте даты с помощью команды ls и убедитесь, что они работают, как ожидалось.



**12. Копирование метки времени последнего доступа к файлу для символьных ссылок**

Ключ **-h** или **-no-dereference** позволяет пользователям копировать время последнего доступа к файлу по символическим ссылкам. Однако, этот ключ будет работать только на системах, которые позволяют изменять время символических ссылок.

$ touch -h link

Эта команда не работает во многих дистрибутивах Linux из-за различных проблем с безопасностью.

**13. Просмотр справочной информации**

Страница справки содержит сводную информацию обо всех возможных вариантах команды. Это очень полезно, так как пользователи могут просто взглянуть на эту страницу и вспомнить, что должен делать тот или иной вариант команды.

$ touch --help

Данная команда покажет страницу справки для linux-команды touch в вашем терминале. Это избавит Вас от необходимости запоминать использование каждого ключа.

**14. Просмотр мануала по команде touch**

Страница man или руководство содержит подробную информацию обо всех доступных вариантах команд терминала Linux. Вы можете просмотреть эту страницу в любое время, выполнив следующую простую команду.

$ man touch

Она покажет вам подробную документацию о том, как работать с командой touch в Linux системах. Обращайтесь к этой документации всякий раз, когда у вас возникнет путаница в отношении использования ключей.

**15. Просмотр информации о версии**

Вы можете просмотреть, какая версия команды touch установлена в вашей системе, с помощью приведенной ниже команды. Она поможет вам определить различную информацию об установленном пакете.

$ touch --version

**Заключительные слова**

Команда touch - одна из самых простых команд терминала Linux. Она очень проста в изучении благодаря ограниченному количеству ключей. Однако, она может быть действительно полезна в ряде ситуаций, когда системные администраторы используют эту команду для управления временными метками файлов.