

tmux /  
tmux

&lt;&gt; Code

Issues 48

Pull requests 4

Discussions

Actions

Wiki

Security



# Getting Started

[Jump to bottom](#)

Nicholas Marriott edited this page 13 октября 2022 г. · 111 revisions

## Getting started

### About tmux

tmux is a program which runs in a terminal and allows multiple other terminal programs to be run inside it. Each program inside tmux gets its own terminal managed by tmux, which can be accessed from the single terminal where tmux is running - this called multiplexing and tmux is a terminal multiplexer.

tmux - and any programs running inside it - may be detached from the terminal where it is running (the outside terminal) and later reattached to the same or another terminal.

Programs run inside tmux may be full screen interactive programs like *vi(1)* or *top(1)*, shells like *bash(1)* or *ksh(1)*, or any other program that can be run in a Unix terminal.

There is a powerful feature set to access, manage and organize programs inside tmux, both interactively and from scripts.

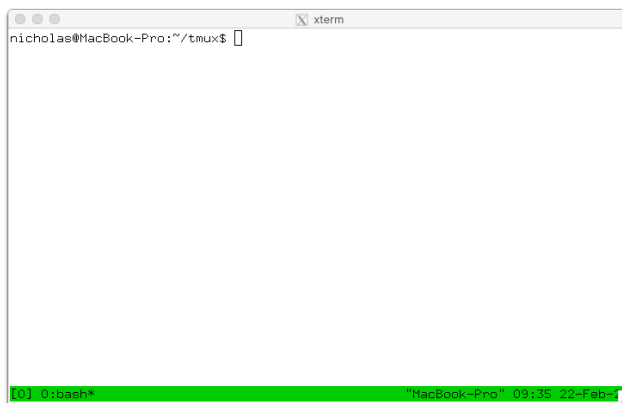
The main uses of tmux are to:

- Protect running programs on a remote server from connection drops by running them inside tmux.
- Allow programs running on a remote server to be accessed from multiple different local computers.
- Work with multiple programs and shells together in one terminal, a bit like a window manager.

For example:

- A user connects to a remote server using *ssh(1)* from an *xterm(1)* on their work computer and run several programs. perhaps an editor, a compiler and a few shells.
- They work with these programs interactively, perhaps start compiling, then close the *xterm(1)* with tmux and go home for the day.
- They are then able to connect to the same remote server from home, attach to tmux, and continue from where they were previously.

Here is a screenshot of tmux in an *xterm(1)* showing the shell:



## About this document

This document gives an overview of some of tmux's key concepts, a description of how to use the main features interactively and some information on basic customization and configuration.

Note that this document may mention features only available in the latest tmux release. Only the latest tmux release is supported. Releases are made approximately every six months.

tmux may be installed from package management systems on most major platforms. See [this document](#) for instructions on how to install tmux or how to build it from source.

## Other documentation and help

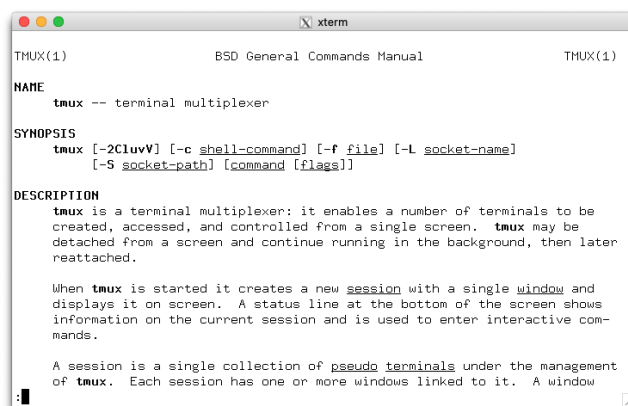
Here are several places to find documentation and help about tmux:

- [The manual page](#) has detailed reference documentation on tmux and a description of every command, flag and option. Once tmux is installed it is also available in section 1:

```
$ man 1 tmux
```



- [The FAQ](#) has solutions to commonly asked questions, mostly about specific configuration issues.
- The [tmux-users@googlegroups.com mailing list](mailto:tmux-users@googlegroups.com).



## Basic concepts

tmux has a set of basic concepts and terms it is important to be familiar with. This section gives a description of how the terminals inside tmux are grouped together and the various terms tmux uses.

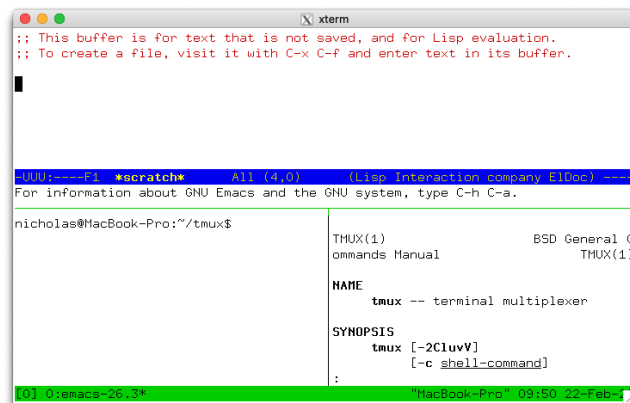
### The tmux server and clients

tmux сохраняет все свое состояние в одном главном процессе, называемом tmux server. Он работает в фоновом режиме и управляет всеми программами, запущенными внутри tmux, и отслеживает их выходные данные. Сервер tmux запускается автоматически, когда пользователь выполняет команду tmux, и по умолчанию завершает работу, когда нет запущенных программ.

Пользователи подключаются к серверу `tmux`, запуская клиент. Он управляет терминалом, на котором запущен, и взаимодействует с сервером, используя файл сокета в `/tmp`. Каждый клиент запускается в одном терминале, который может быть терминалом `X(7)`, таким как `xterm(1)`, системной консолью или терминалом внутри другой программы (например, самого `tmux`). Каждый клиент идентифицируется по имени внешнего терминала, на котором он запущен, например `/dev/ttypf`.

### Сеансы, окна и панели

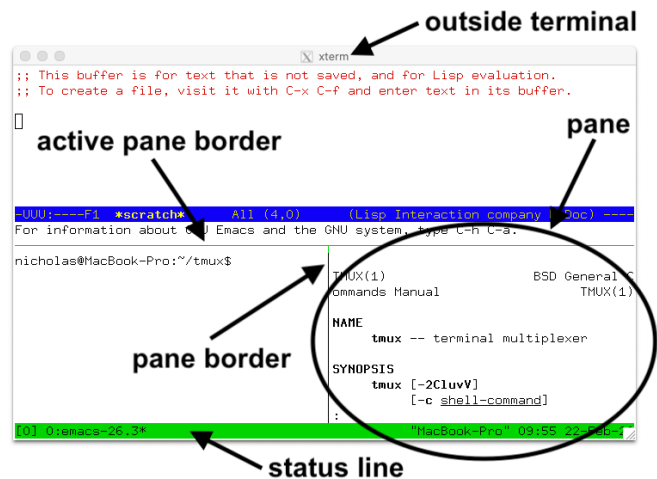
Каждый терминал внутри `tmux` принадлежит одной панели, это прямоугольная область, которая показывает содержимое терминала внутри `tmux`. Поскольку каждый терминал внутри `tmux` отображается только на одной панели, термин `pane` может использоваться для обозначения всей панели, терминала и программы, запущенной внутри него. На скриншоте справа показан `tmux` с панелями.



Каждая панель отображается в одном окне. Окно состоит из одной или нескольких панелей, которые вместе занимают всю его площадь, поэтому несколько панелей могут быть видны одновременно. Окно обычно занимает весь терминал, к которому подключен `tmux`, но оно может быть больше или меньше. Размеры и положения всех панелей в окне называются макетом окна.

У каждого окна есть имя - по умолчанию `tmux` выберет одно, но оно может быть изменено пользователем. Названия окон не обязательно должны быть уникальными, окна обычно идентифицируются по сеансу и индексу окна, а не по их названию.

Каждая панель отделена от окружающих ее панелей линией, это называется границей панели. В каждом окне есть одна панель, называемая активной панелью, именно сюда отправляется любой введенный текст, и это панель по умолчанию, используемая для команд, нацеленных на окно. Граница активной панели отмечена зеленым цветом, или, если панелей всего две, верхняя, нижняя, левая или правая половина границы окрашена в зеленый цвет.



Несколько окон сгруппированы в сеансы. Если окно является частью сеанса, считается, что оно связано с этим сеансом. Окна могут быть связаны с несколькими сеансами одновременно, хотя чаще всего они находятся только в одном. У каждого окна в сеансе есть номер, называемый индексом окна - одно и то же окно может быть связано с разными индексами в разных сеансах. Список окон сеанса - это все окна, связанные с этим сеансом, в порядке их индексов.

У каждого сеанса есть одно текущее окно, это окно отображается при подключении сеанса и является окном по умолчанию для любых команд, нацеленных на сеанс. Если текущее окно изменено, предыдущее текущее окно становится известным как последнее окно.

Сеанс может быть присоединен к одному или нескольким клиентам, что означает, что он отображается на внешнем терминале, где запущен этот клиент. Любой текст, введенный в этот внешний терминал, отправляется на активную панель в текущем окне подключенного сеанса. Сеансы не имеют индекса, но у них есть имя, которое должно быть уникальным.

Краткое описание:

- Программы запускаются в терминалах на панелях, каждая из которых принадлежит одному окну.
- У каждого окна есть название и одна активная панель.
- Окна связаны с одним или несколькими сеансами.
- Каждый сеанс содержит список окон, каждое из которых имеет индекс.
- Одним из окон сеанса является текущее окно.
- Сеансы подключены к одному или нескольким клиентам или отсоединены (не подключены ни к каким клиентам).
- Каждый клиент привязан к одному сеансу.

Краткое изложение терминов

Термин	Описание
Клиент	Подключает сеанс <code>tmux</code> с внешнего терминала, такого как <code>xterm(1)</code>
Сессия	Объединяет одно или несколько окон вместе
Окно	Объединяет одну или несколько панелей вместе, связанных с одним или несколькими сеансами
Панель	Содержит терминал и запущенную программу, отображается в одном окне
Активная панель	Панель в текущем окне, на которую отправляется ввод текста; по одной на каждое окно
Текущее окно	Окно в прикрепленном сеансе, куда отправляется ввод текста; по одному на сеанс
Последнее окно	Предыдущее текущее окно
Название сеанса	В названии сеанса по умолчанию используется число, начинающееся с нуля
Список окон	Список окон в сеансе в порядке нумерации
Название окна	Название окна по умолчанию соответствует имени запущенной программы на активной панели
Индекс окна	Номер окна в списке окон сеанса
Макет окна	Размер и расположение панелей в окне

Интерактивное использование `tmux`

Создание сеансов

Чтобы создать первый сеанс `tmux`, `tmux` запускается из командной строки. Новый сеанс создается с помощью `new-session` команды - `new` для краткости:

```
$ tmux new
```



Без аргументов, `new-session` создает новый сеанс и присоединяет его. Поскольку это первый сеанс, сервер `tmux` запускается, и `tmux` запускается из командной строки `shell` становится первым клиентом и подключается к нему.

Новый сеанс будет иметь одно окно (с индексом 0) с единственной панелью, содержащей оболочку. Приглашение командной строки должно появиться в верхней части терминала, а зеленая строка состояния - внизу (подробнее о строке состояния ниже).

По умолчанию будет вызываться первый сеанс 0, второй 1 и так далее. `new-session` позволяет указать имя сеанса с помощью `-s` флага:

```
$ tmux new -smysession
```



Это создает новый сеанс с именем `mysession`. Вместо этого может быть дана команда вместо запуска командной оболочки путем передачи дополнительных аргументов. Если задан один аргумент, `tmux` передаст его командной оболочке, если их несколько, она запустит команду напрямую. Например, они запускают `emacs(1)`:

```
$ tmux new 'emacs ~/.tmux.conf'
```



Или:

```
$ tmux new -- emacs ~/.tmux.conf
```



По умолчанию `tmux` вызывает первое окно в сеансе после всего, что в нем запущено. Флаг `-n` указывает имя для использования вместо него, в данном случае это окно `mytopwindow` запущенное *сверху* (1):

```
$ tmux new -nmytopwindow top
```

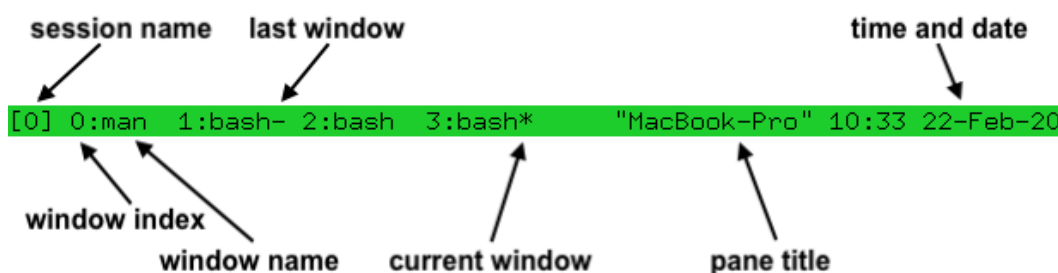


`new-session` имеет другие флаги - некоторые описаны ниже. Полный список есть [в руководстве по tmux](#).

## Строка состояния

Когда клиент `tmux` подключен, в нижней строке экрана отображается строка состояния. По умолчанию она зеленая и показывает:

- Слева указано название прикрепленного сеанса: `[0]`.
- В середине приведен список окон сеанса с их индексом, например, с одним окном, вызываемым `ksh` с индексом 0: `0:ksh`.
- Справа заголовок панели в кавычках (по умолчанию это имя хоста, на котором запущен `tmux`), а также время и дата.



По мере открытия новых окон список окон растет - если их слишком много, чтобы поместиться по ширине терминала, слева или справа или в обоих местах будет добавлено `< ог >`, чтобы показать наличие скрытых окон.

В списке окон текущее окно помечено символом `*` после названия, а последнее окно - символом `-`.

### Префиксный ключ

После подключения клиента `tmux` все введенные ключи пересылаются программе, запущенной в активной области текущего окна. Для клавиш, управляющих самим `tmux`, сначала необходимо нажать специальную клавишу - она называется префиксной клавишей.

По умолчанию используется префиксный ключ `C-b`, что означает `Ctrl` ключ и `b`. В `tmux` клавиши-модификаторы отображаются с помощью префикса `C-` для клавиши управления, `M-` для мета-клавиши (обычно `Alt` на современных компьютерах) и `S-` для клавиши `shift`. Они могут быть объединены вместе, поэтому `C-M-x` означает совместное нажатие клавиши `control`, клавиши `meta` и `x`.

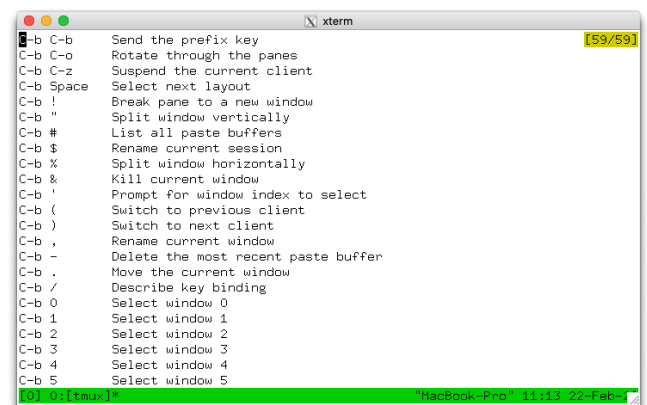
Когда нажата префиксная клавиша, `tmux` ожидает следующего нажатия клавиши, и это определяет, какая команда `tmux` будет выполнена. Здесь показаны подобные клавиши с пробелом между ними: `C-b c` означает, что сначала нажимается префиксная клавиша `C-b`, затем она отпускается, а затем нажимается `c` клавиша. Необходимо соблюдать осторожность, чтобы отпустить `Ctrl` клавишу после нажатия `C-b`, если это необходимо - `C-b c` отличается от `C-b C-c`.

Нажатие `C-b` дважды отправляет `C-b` клавишу программе, запущенной на активной панели.

### Ключи справки

Каждая привязка ключа `tmux` по умолчанию имеет краткое описание, помогающее запомнить, что делает ключ. Список всех ключей можно просмотреть, нажав `C-b ?`.

`C-b ?` переходит в режим просмотра для отображения текста. Панель в режиме просмотра имеет свои собственные привязки клавиш, для которых не требуется префиксный ключ. В целом они соответствуют *emacs(1)*. Наиболее важными являются `Up`, `Down`, `C-Up`, `C-Down` прокрутка вверх и вниз и `q` выход из режима. Номер строки в верхней видимой строке вместе с общим количеством строк показан в правом верхнем углу.



В качестве альтернативы, тот же список можно просмотреть из командной строки, выполнив:

```
$ tmux ls -N|more
```



`C-b /` показывает описание отдельной клавиши - внизу появится подсказка терминала. При нажатии клавиши ее описание отобразится в том же месте. Например, при нажатии `C-b /` затем `?` отображается:

```
C-b ? List key bindings
```



### Команды и флаги

tmux имеет большой набор команд. Все они имеют название, подобное `new-window` or `new-session` или `list-keys`, и у многих также есть более короткий псевдоним, такой как `neww` or `new` или `lsk`.

Каждый раз, когда используется привязка ключа, выполняется одна или несколько команд tmux. Например, `C-b c` выполняется `new-window` команда.

Команды также можно использовать из командной оболочки, как с `new-session` и `list-keys` выше.

Каждая команда имеет ноль или более флагов, точно так же, как и стандартные команды Unix. Флаги могут сами принимать или не принимать один аргумент. Кроме того, команды могут принимать дополнительные аргументы после флагов. Флаги передаются после команды, например, для запуска `new-session` команды (псевдонима `new`) с флагами `-d` и `-n`:

```
$ tmux new-session -d -nmysession
```



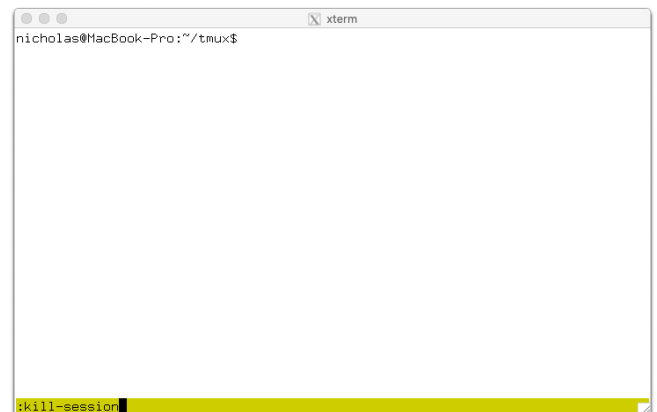
Все команды и их флаги задокументированы на странице руководства по tmux.

В этом документе основное внимание уделяется доступным привязкам клавиш, но команды упоминаются для информации или там, где есть полезный флажок. Их можно ввести из командной строки, описанной в следующем разделе.

## Командная строка

в tmux есть интерактивная командная строка. Ее можно открыть нажатием `C-b :` и она отображается вместо строки состояния, как показано на этом скриншоте.

В командной строке команды можно вводить аналогично тому, как они вводятся в командной оболочке. Выходные данные либо будут отображаться в течение короткого периода времени в строке состояния, либо переключат активную панель в режим просмотра.



По умолчанию в командной строке используются ключи, аналогичные `emacs(1)`; однако, если для переменных окружения `VISUAL` or `EDITOR` задано значение, содержащее что-либо `vi` (например, `vi` or `vim` или `nvi`), то вместо них используются ключи в стиле `vi(1)`.

Несколько команд можно вводить вместе в командной строке, разделяя их точкой с запятой ( `;` ). Это называется последовательностью команд.

## Подключение и отсоединение

Отключение от tmux означает, что клиент завершает работу и отсоединяется от внешнего терминала, возвращаясь в оболочку и оставляя сеанс tmux и любые программы внутри него запущенными в фоновом режиме. Для отключения tmux используется `C-b d` привязка ключа. Когда tmux отключится, будет напечатано сообщение с названием сеанса:

```
[detached (from session mysession)]
```



В `attach-session` командование прикрепляет к существующему сеансу. Без аргументы, он будет подключаться к наиболее часто используемых сессии, что уже не прикреплен:

```
$ tmux attach
```



Или `-t` дает название сеанса, к которому нужно подключиться:

```
$ tmux attach -tmysession
```



По умолчанию, подключение к сессии не отсоединяет никакие другие клиенты не прилагается на той же сессии. Этот `-d` флаг делает это:

```
$ tmux attach -dmysession
```



У `new-session` команды есть `-A` флаг, который нужно прикрепить к существующему сеансу, если он существует, или создать новый, если его нет. Для сеанса с именем `mysession`:

```
$ tmux new -Amysession
```



Флаг `-D` может быть добавлен, чтобы заставить `new-session` также вести себя как `attach-session` с `-d` и отсоединять любые другие клиенты, подключенные к сеансу.

## Список сеансов

`list-session` Команда (псевдоним `ls`) показывает список доступных сеансов, которые можно подключить. Здесь показаны четыре сеанса, которые называются `1`, `2`, `myothersession` и `mysession`:

```
$ tmux ls
1: 3 windows (created Sat Feb 22 11:44:51 2020)
2: 1 windows (created Sat Feb 22 11:44:51 2020)
myothersession: 2 windows (created Sat Feb 22 11:44:51 2020)
mysession: 1 windows (created Sat Feb 22 11:44:51 2020)
```



## Полностью уничтожаем tmux

Если внутри `tmux` нет сеансов, окон или панелей, сервер завершит работу. Его также можно полностью отключить с помощью `kill-server` команды. Например, в командной строке:

```
:kill-server
```



## Создание новых окон

Новое окно может быть создано в подключенном сеансе с помощью `C-b` с привязки ключа, который запускает `new-window` команду. Новое окно создается по первому доступному индексу, поэтому второе окно будет иметь индекс `1`. Новое окно становится текущим окном сеанса.

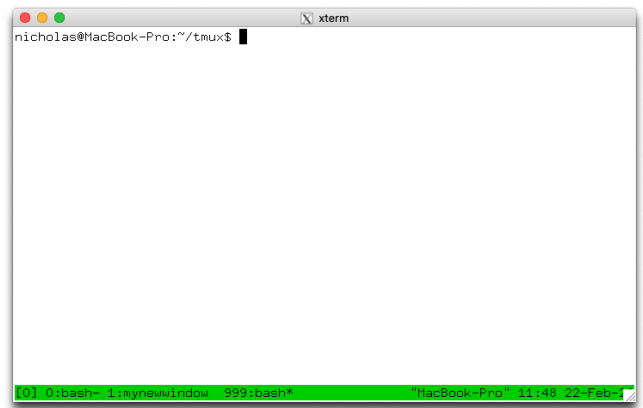
Если в списке окон есть какие-либо пробелы, они заполняются новыми окнами. Таким образом, если есть окна с индексами `0` и `2`, следующее новое окно будет создано с индексом `1`.

Команда `new-window` имеет несколько полезных флагов, которые можно использовать в командной строке:



- Флаг `-d` создает окно, но не делает его текущим.
- `-n` позволяет задать имя для нового окна. Например, с помощью командной строки создать окно с именем `mynewwindow`, не делая его текущим окном.:

```
:neww -dnmynewwindow
```



- Флаг `-t` указывает цель для окна. Цели команды имеют особый синтаксис, но для простого использования с `new-window` достаточно просто указать индекс окна. При этом создается окно с индексом 999:

```
:neww -t999
```



Команда для запуска в новом окне может быть передана `new-window` тем же способом, что и `new-session`. Например, для создания нового окна под управлением `top(1)`:

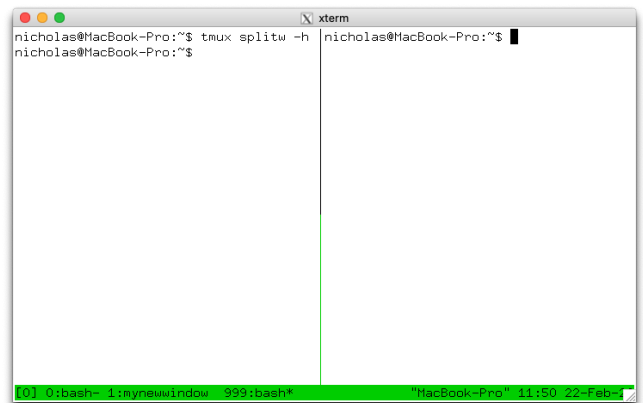
```
:neww top
```



## Разделение окна

Панель создается путем разделения окна. Это делается с помощью `split-window` команды, которая по умолчанию привязана к двум клавишам:

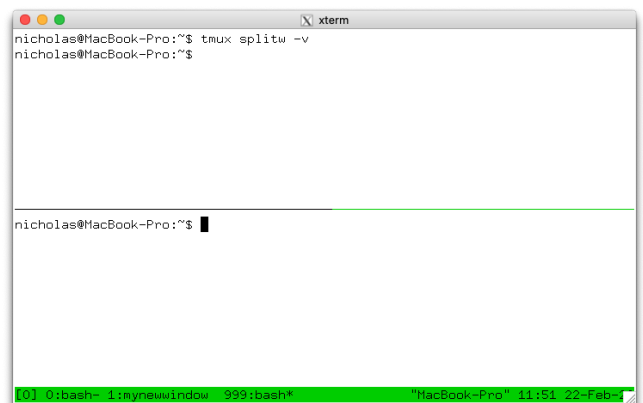
- `C-b %` разбивает текущую панель на две по горизонтали, создавая две панели рядом друг с другом, одну слева и одну справа.
- `C-b "` разбивает текущую панель на две по вертикали, создавая две панели, расположенные одна над другой.



Каждый раз, когда панель разделяется на две, каждую из этих панелей можно разделить снова, используя те же привязки клавиш, пока панель не станет слишком маленькой.

`split-window` имеет несколько полезных флагов:

- `-h` выполняет горизонтальное разделение и `-v` вертикальное разделение.
- `-d` активная панель не изменяется на вновь созданную.
- `-f` создает новую панель, охватывающую всю ширину или высоту окна, вместо того, чтобы ограничиваться размером разделяемой панели.



- `-b` новая панель размещается слева или выше разделяемой панели, а не справа или ниже.

Команда для запуска в новой панели может быть передана `split-window` таким же образом, как `new-session` и `new-window`.

### Изменение текущего окна

Существует несколько привязок клавиш для изменения текущего окна сеанса:

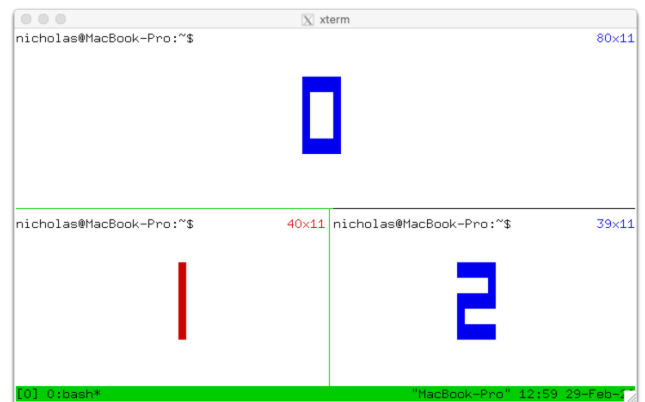
- `C-b 0` изменения в window 0, `C-b 1` в window 1, вплоть до `C-b 9` для window 9.
- `C-b '` запрашивает индекс окна и изменения в этом окне.
- `C-b n` переход к следующему окну в списке окон по номеру. Таким образом, нажатие `C-b n` в окне 1 приведет к переходу в окно 2, если оно существует.
- `C-b p` изменения предыдущего окна в списке окон по номеру.
- `C-b l` изменения в последнем окне, то есть окне, которое было последним по сравнению с текущим окном перед текущим окном.

Все это варианты команды `select-window`.

### Изменение активной панели

Активную панель можно переключать между панелями в окне с помощью следующих привязок клавиш:

- `C-b Up`, `C-b Down`, `C-b Left` и `C-b Right` перейдите на панель выше, ниже, слева или справа от активной панели. Эти клавиши охватывают окно, поэтому нажатие `C-b Down` на панель внизу приведет к появлению панели сверху.
- `C-b q` номера панелей и их размеры на короткое время выводятся поверх панелей. Нажатие одной из цифровых клавиш перед тем, как они исчезнут, изменит активную панель на выбранную, поэтому `C-b q 1` изменится на панель под номером 1.
- `C-b o` переходит к следующей панели по номеру панели и `C-b C-o` заменяет эту панель на активную, чтобы они поменялись местами и размерами в окне.



В них используются команды `select-pane` и `display-panes`.

Номера панелей не фиксированы, вместо этого панели нумеруются по их положению в окне, поэтому, если панель с номером 0 заменяется панелью с номером 1, номера меняются местами так же, как и сами панели.

### Выбор сеансов, окон и панелей

tmux включает режим, в котором сеансы, окна или панели могут быть выбраны из дерева, это называется режимом дерева. Его можно использовать для просмотра сеансов, окон и панелей; для изменения подключенного сеанса, текущего окна или активной панели; для отключения сеансов, окон и панелей; или применить команду к нескольким сразу, пометив их.

Для перехода в древовидный режим есть две привязки клавиш: C-b s начинает показывать только сеансы и с выбранным подключенным сеансом; C-b w начинается с сеансов, развернутых таким образом, чтобы отображались окна, и с выбранным текущим окном в подключенном сеансе.

Древовидный режим разбивает окно на две части: верхняя половина содержит дерево сеансов, окон и панелей, а нижняя половина содержит предварительный просмотр области вокруг курсора на каждой панели. Для сеансов предварительный просмотр показывает активные панели в таком количестве окон, которое поместится; для Windows - столько панелей, сколько поместится; а для панелей - только выбранную панель.

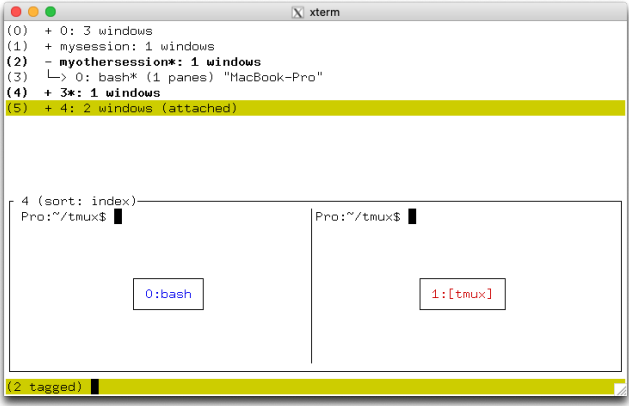
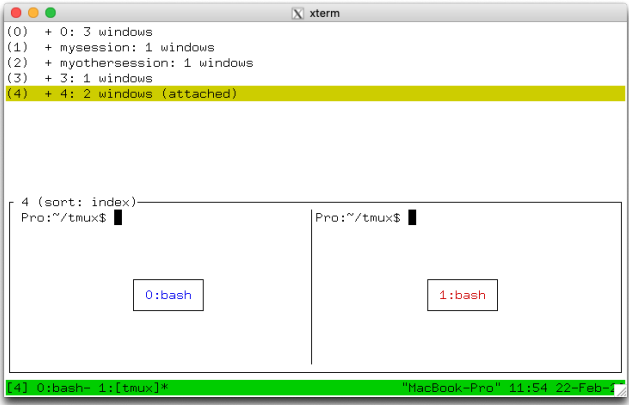
Для клавиш управления древовидным режимом не требуется префикс. Для навигации по списку можно использовать клавиши Up и Down . Enter изменение выбранного элемента (он становится подключенным сеансом, текущим окном или активной панелью) и выход из режима. Right расширяет элемент, если это возможно - сеансы расширяются, чтобы показать их окна, а windows - их области. Left сворачивает элемент, чтобы скрыть все окна или области. O изменяет порядок элементов и q выходит из режима дерева.

Элементы в дереве помечаются нажатием t и снимаются с пометок повторным нажатием t . Помеченные элементы выделены жирным шрифтом и с \* после их названия. Все помеченные элементы могут быть отключены нажатием T . Помеченные элементы могут быть уничтожены вместе нажатием X или команда, примененная ко всем из них нажатием : для получения подсказки.

Каждый элемент в дереве имеет комбинацию клавиш as в скобках в начале строки. Нажатие этой клавиши немедленно приведет к выбору этого элемента (как если бы он был выбран и Enter нажат). Первые десять элементов являются ключами 0 к 9 , а после этого используются ключи M-a к M-z .

Это список ключей, доступных в древовидном режиме без нажатия префиксной клавиши:

Клавиша	Функция
Enter	Измените подключенный сеанс, текущее окно или активную панель
Up	Выберите предыдущий пункт
Down	Выберите следующий пункт
Right	Развернуть статью
Left	Свернуть элемент



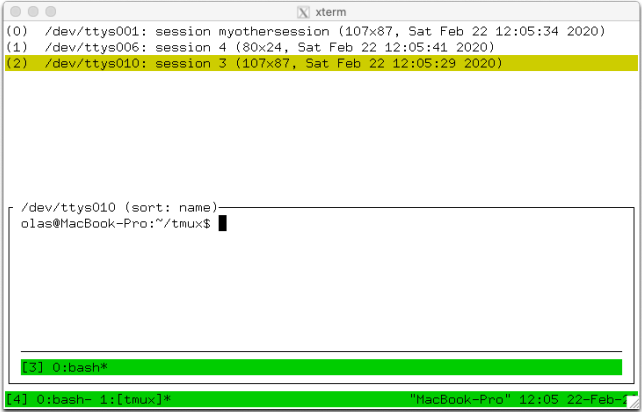
Клавиша	Функция
x	Уничтожить выбранный предмет
X	Уничтожать объекты с тегами
<	Прокрутите предварительный просмотр влево
>	Прокрутите предварительный просмотр вправо
C-s	Поиск по названию
n	Повторить последний поиск
t	Переключать, если элемент помечен
T	Отмечать элементы нельзя
C-t	Отмечайте все элементы тегами
:	Запрос команды для запуска для выбранного элемента или каждого помеченного элемента
O	Измените поле сортировки
r	Обратный порядок сортировки
v	Переключить предварительный просмотр
q	Выход из режима дерева

Режим дерева активируется командой `choose-tree` .

Отключение других клиентов

Список клиентов доступен нажатием `C-b D` (то есть `C-b S-d` ). Это похоже на древовидный режим и называется клиентским режимом.

Каждый клиент показан в списке в верхней половине с указанием его имени, подключенного сеанса, размера, а также времени и даты последнего использования; в нижней половине представлен предварительный просмотр выбранного клиента с указанием той части строки состояния, которая поместится.



Клавиши перемещения и метки такие же, как в режиме дерева, но другие отличаются, например, `Enter` клавиша отсоединяет выбранного клиента.

Это список ключей в клиентском режиме без перемещения и пометки ключей, которые такие же, как в древовидном режиме:

Клавиша	Функция
Enter	Отсоединить выбранный клиент
d	Отсоедините выбранный клиент, так же, как <code>Enter</code>
D	Отключать помеченные клиенты
x	Отключите выбранный клиент и попробуйте отключить оболочку, из которой он был запущен

Клавиша	Функция
X	Отключите помеченные клиенты и попытайтесь уничтожить оболочки, с которых они были запущены

Помимо использования клиентского режима, у `detach-client` команды есть `-a` флаг для отключения всех клиентов, кроме подключенного клиента.

## Завершение сеанса, окна или панели

Нажатие C-b & запрашивает подтверждение, после чего текущее окно отключается. Все панели в окне отключаются одновременно. C-b x отключается только активная панель. Они привязаны к командам kill-window и kill-pane.

Команда `kill-session` завершает работу подключенного сеанса и всех его окон и отключает клиент. Привязки ключа для `kill-session` нет, но его можно использовать из командной строки или :  
командной строки в древовидном режиме.

## Переименование сеансов и окон

C-b \$ запросит новое имя для подключенного сеанса. Для этого используется `rename-session` команда. Аналогично, C-b , запрашивает новое имя для текущего окна с помощью `rename-window` команды.

## Замена и перемещение

tmux позволяет менять местами панели и окна с помощью команд `swap-pane` и `swap-window`.

Для упрощения обмена можно пометить одну панель. Во всех сеансах есть одна отмеченная панель. C-b m Привязка ключа определяет, является ли активная область в текущем окне подключенного сеанса отмеченной областью. C-b M полностью очищает отмеченную область, так что никакая область не отмечена. Отмеченная область отображается зеленым фоном по краям, а окно, содержащее отмеченную область, имеет M флаг в строке состояния.

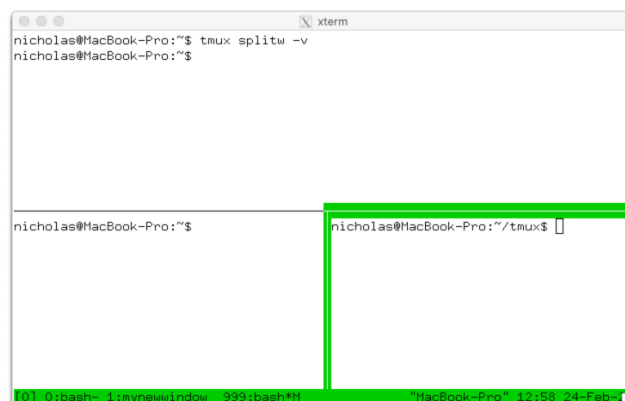
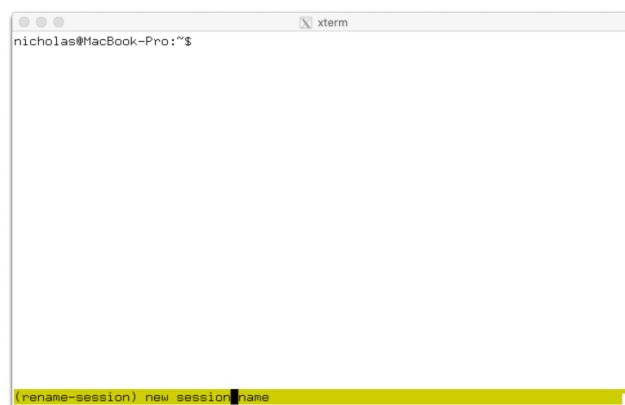
После того, как панель помечена, ее можно заменить активной панелью в текущем окне с помощью `swar-pane` команды, или окно, содержащее отмеченную панель, можно заменить текущим окном с помощью `swap-window` команды. Например, с помощью командной строки:

```
: swap-pane
```



Панели дополнительно можно менять местами с панелью выше или ниже, используя привязки клавиш C-b { и C-b } .

Для перемещения окон используется `move-window` команда или `C-b .` привязка клавиш. Нажатие `C-b .` запросит новый индекс для текущего окна. Если окно с заданным индексом уже существует, будет показана ошибка. Существующее окно можно заменить, используя флаг `-k` - для перемещения окна с индексом 999:



```
:move-window -kt999
```



Если в списке окон есть пробелы, индексы можно перенумеровать с помощью флага `-r` на `move-window` . Например, это изменит список окон из `0, 1, 3, 999` на `0, 1, 2, 3`:

```
:movew -r
```



Изменение размера и масштабирования панелей

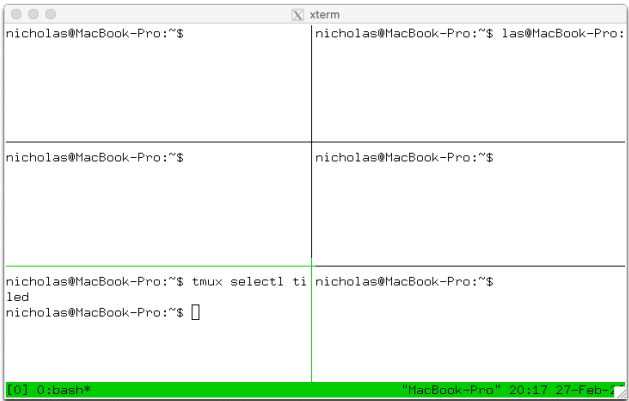
Размеры панелей можно изменять небольшими шагами с помощью `C-b C-Left` , `C-b C-Right` , `C-b C-Up` и `C-b C-Down` и большими шагами с помощью `C-b M-Left` , `C-b M-Right` , `C-b M-Up` и `C-b M-Down` . В них используется команда `resize-pane` .

Можно временно сделать так, чтобы одна панель занимала все окно вместе с `C-b z` , скрывая любые другие панели. Повторное нажатие `C-b z` возвращает панель и расположение окна к тому, что было. Это называется масштабированием и разархивированием. Окно, в котором была увеличена панель, помечено `z` в строке состояния. Команды, которые изменяют размер или положение панелей в окне, автоматически уменьшают масштаб окна.

Макеты окон

Панели в окне могут автоматически располагаться в одном из нескольких именованных макетов, их можно переключать с помощью `C-b Space` привязки клавиш или выбирать непосредственно с помощью `C-b M-1` , `C-b M-2` и так далее.

Доступные макеты следующие:

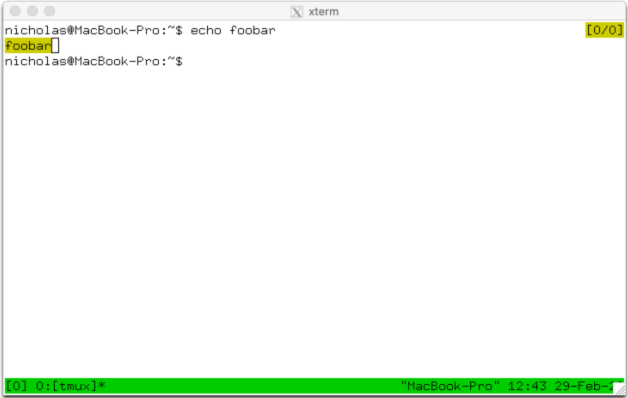


Имя	Клавиша	Описание
четный-горизонтальный	C-b M-1	Равномерно распределить по
четный-вертикальный	C-b M-2	Равномерно распределите вверх и вниз
главная-горизонтальная	C-b M-3	Одна большая панель сверху, остальные равномерно распределены по
главная-вертикальная	C-b M-4	Одна большая панель слева, остальные равномерно распределены вверх и вниз
выложенный плиткой	C-b M-5	Выложено плиткой с тем же количеством строк, что и столбцов

Скопируйте и вставьте

в `tmux` есть собственная система копирования и вставки. Фрагмент скопированного текста называется буфером вставки. Текст копируется с помощью режима копирования, вводится с помощью `C-b [`, а самый последний скопированный текст вставляется в активную панель с помощью `C-b ]`.

Буферам вставки можно давать имена, но по умолчанию `tmux` присваивает им имя, например `buffer0` или `buffer1`. Буферы, подобные этому, называются автоматическими, и их сохраняется не более 50 - если имеется 50 буферов, самый старый удаляется при добавлении другого. Если буферу присвоено имя, он называется именованным буфером; именованные буферы не удаляются, независимо от того, сколько их существует.



Можно настроить `tmux` для отправки любого скопированного текста в системный буфер обмена: [в этом документе](#) объясняются различные способы настройки этого.

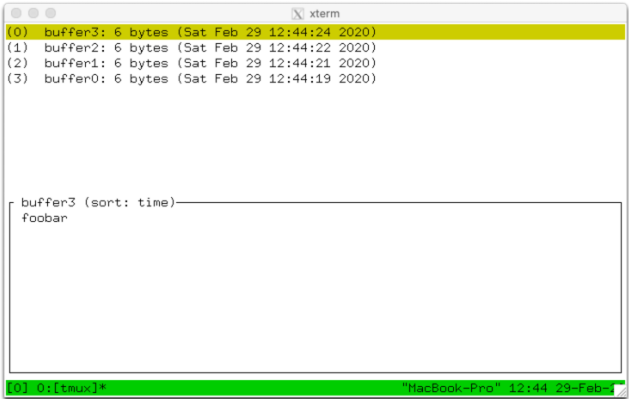
Режим копирования замораживает любой вывод на панели и позволяет копировать текст. Режим просмотра (описанный ранее) - это форма режима копирования, доступная только для чтения.

Как и в командной строке, в режиме копирования используются ключи, аналогичные `emacs(1)`; однако, если для переменных окружения `VISUAL` или `EDITOR` установлено значение, содержащее что-либо `vi`, то вместо них используются ключи в стиле `vi(1)`. Следующие ключи - некоторые из тех, которые доступны в режиме копирования с помощью ключей `emacs(1)`:

Клавиша	Экшен
Up , Down , Left , Right	Переместите курсор
C-Space	Начать выбор
C-w	Скопируйте выделенное и выйдите из режима копирования
q	Выйдите из режима копирования
C-g	Остановите выбор без копирования или прекратите поиск
C-a	Переместите курсор в начало строки
C-e	Переместите курсор в конец строки
C-r	Интерактивный поиск в обратном направлении
M-f	Переместите курсор к следующему слову
M-b	Переместите курсор на предыдущее слово

Полный список ключей для `vi(1)` и `emacs(1)` [доступен на странице руководства](#).

После копирования некоторого текста можно вставить самый свежий с помощью `C-b ]` или вставить более старый буфер с помощью буферного режима, введенного с помощью `C-b =`. Режим буфера аналогичен режиму клиента и режиму дерева и предлагает список буферов вместе с предварительным просмотром их содержимого. Помимо клавиш навигации и тегов, используемых в режиме дерева и клиентском режиме, буферный режим поддерживает следующие клавиши:



Клавиша	Функция
Enter	Вставить выбранный буфер
p	Вставьте выбранный буфер, так же, как Enter
P	Вставка буферов с тегами
d	Удалить выбранный буфер
D	Удаление помеченных буферов

Буфер можно переименовать с помощью `set-buffer` команды. Флаг `-b` присваивает существующее имя буфера и `-n` новое имя. Это преобразует его в именованный буфер. Например, чтобы переименовать `buffer0` в `mybuffer` из командной строки:

```
:setb -bbuffer0 -nmybuffer
```

`set-buffer` также может использоваться для создания буферов. Для создания буфера с именем `foo` с текстом `bar` :

```
:setb -bfoo bar
```

`load-buffer` загрузит буфер из файла:

```
:loadb -bbuffername ~/a/file
```

`set-buffer` или `load-buffer` без `-b` создает автоматический буфер.

Существующий буфер можно сохранить в файл с помощью `save-buffer` :

```
:saveb -bbuffer0 ~/saved_buffer
```

Поиск окон и панелей

`C-b f` запрашивает некоторый текст, а затем переходит в режим дерева с фильтром для отображения только тех областей, где этот текст отображается в видимом содержимом или заголовке области или в названии окна. Если панели найдены, в дереве отображаются только эти панели, а текст `filter: active` отображается над предварительным просмотром. Если панели не найдены, все панели отображаются в виде дерева, а текст `filter: no matches` отображается над предварительным просмотром.



С помощью мыши

tmux имеет расширенную поддержку мыши. Его можно использовать для изменения активной области или окна, изменения размера панелей, копирования текста или выбора элементов в меню.

Поддержка мыши включена с помощью опции `mouse`; опции и файл конфигурации подробно описаны в следующем разделе. Чтобы включить мышь из командной строки, используйте команду `set-option`:

```
:set -g mouse on
```



После включения мыши:

- Нажатие левой кнопки на панели делает эту панель активной.
- Нажатие левой кнопки на название окна в строке состояния делает его текущим окном.
- Перетаскивание левой кнопкой мыши границы панели изменяет размер панели.
- При перетаскивании левой кнопкой мыши внутри панели выделяется текст; выделенный текст копируется при отпускании мыши.
- Нажатие правой кнопки на панели открывает меню с различными командами. Когда отпущена кнопка мыши, выбранная команда запускается с панелью в качестве целевой. У каждого пункта меню также есть сочетание клавиш, указанное в скобках.
- Нажатие правой кнопки в окне или на название сеанса в строке состояния открывает аналогичное меню для окна или сеанса.

## Настройка tmux

### Файл конфигурации

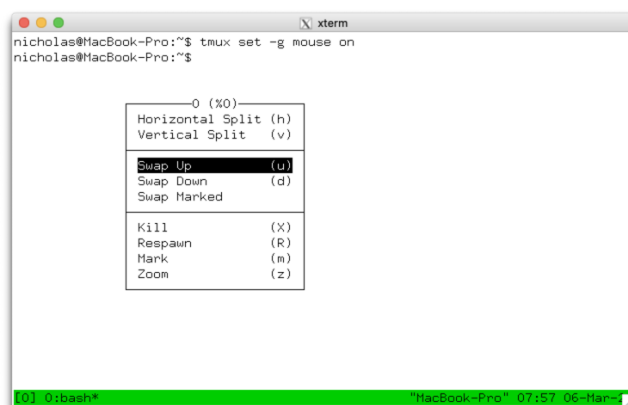
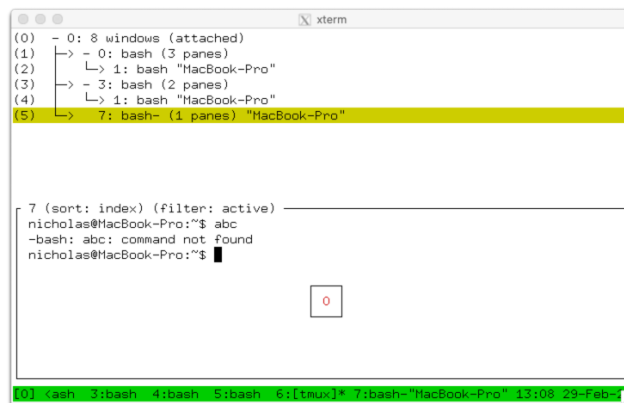
При запуске сервера `tmux` `tmux` запускает файл с именем `.tmux.conf` в домашнем каталоге пользователя. Этот файл содержит список команд `tmux`, которые выполняются по порядку. Важно отметить, что `.tmux.conf` это *только* запускается при запуске сервера, а не при создании нового сеанса.

Другой файл конфигурации может быть запущен с `.tmux.conf` или с работающего сервера `tmux` с помощью `source-file` команды, например, для повторного запуска `.tmux.conf` с работающего сервера с помощью командной строки:

```
:source ~/.tmux.conf
```



Команды в файле конфигурации отображаются по одной в строке. Любые строки, начинающиеся с `#`, являются комментариями и игнорируются:



```
# This is a comment - the command below turns the status line off
set -g status off
```



Строки в файле конфигурации обрабатываются аналогично командной строке, например:

- Аргументы могут быть заключены в ' или " включать пробелы, или пробелы могут быть экранированы. Эти четыре строки выполняют то же самое.:

```
set -g status-left "hello word"
set -g status-left "hello\ word"
set -g status-left 'hello word'
set -g status-left hello\ word
```



- Но экранирование не происходит внутри ' s . Строка здесь hello\ world не hello world :

```
set -g status-left 'hello\ word'
```



- ~ расширен до домашнего каталога (за исключением внутренних ' каталогов):

```
source ~/myfile
```



- Переменные среды могут быть установлены, а также расширены (но не внутри ' ов):

```
MYFILE=myfile
source "~/MYFILE"
```



Любые переменные, заданные в файле конфигурации, будут переданы в новые панели созданные внутри tmux.

- Заменены несколько специальных символов, таких как \n (перевод строки) и \t (табуляция). Литерал \ должен быть указан как \\ .

Хотя файлы конфигурации tmux обладают некоторыми функциями, аналогичными shell, они не являются сценариями оболочки и не могут использовать конструкции оболочки, подобные \$( ) .

## Привязки клавиш

привязки ключей tmux изменяются с помощью команд bind-key и unbind-key . Каждая привязка ключей в tmux принадлежит именованной таблице ключей. Существует четыре таблицы ключей по умолчанию:

- В root таблице приведены привязки клавиш для нажатых клавиш без префикса key.
- В prefix таблице приведены привязки клавиш для клавиш, нажимаемых после префиксной клавиши, подобные тем, которые упоминались до сих пор в этом документе.
- В copy-mode таблице приведены привязки клавиш для ключей, используемых в режиме копирования с помощью ключей в стиле *emacs(1)*.

- В `copy-mode-vi` таблице приведены привязки клавиш для клавиш, используемых в режиме копирования с помощью клавиш в стиле *vi(1)*.

Все привязки клавиш или те, которые относятся к одной таблице, могут быть перечислены с помощью `list-keys` команды. По умолчанию клавиши отображаются в виде серии `bind-key` команд. Флаг `-T` указывает таблицу ключей для отображения, а флаг `-N` показывает справку по ключу, такую как `C-b ?` привязка ключа.

Например, чтобы перечислить только ключи в `prefix` таблице:

```
$ tmux lsk -T prefix
bind-key -T prefix C-b send-prefix
bind-key -T prefix C-o rotate-window
...
```



Или:

```
$ tmux lsk -T prefix -N
C-b Send the prefix key
C-o Rotate through the panes
...
```



`bind-key` команды могут использоваться для установки привязки ключа либо в интерактивном режиме, либо чаще всего из файла конфигурации. Как и `list-keys`, `bind-key` имеет флаг `-T` для использования таблицы ключей. Если `-T` не указан, ключ помещается в `prefix` таблицу; `-n` флаг является сокращением для `-T root` использования `root` таблицы.

Например, `list-keys` команда показывает, что `C-b 9` изменения в окне 9 выполняются с помощью `select-window` команды:

```
$ tmux lsk -T prefix 9
bind-key -T prefix 9 select-window -t :=9
```



Аналогичную привязку ключа для внесения `C-b M-0` изменений в `window 10` можно добавить следующим образом:

```
bind M-0 selectw -t:=10
```



`-t` Флаг `to select-window` указывает целевое окно. В этом примере, `:` означает, что целью является окно и `=` означает, что название должно точно совпадать `10`. Цели более подробно описаны в [разделе КОМАНД](#) [страницы руководства](#).

Команда `unbind-key` удаляет привязку ключа. Например, у `bind-key` нее есть флаги `-T` и `-n` для таблицы ключей. Нет необходимости удалять привязку ключа перед повторной привязкой, `bind-key` заменит любую существующую привязку ключа. `unbind-key` необходимо только полностью удалить привязку ключа:

```
unbind M-0
```



Привязки клавиш режима копирования

Привязки клавиш режима копирования задаются в `copy-mode` и `copy-mode-vi` ключевых таблицах. Режим копирования имеет отдельный набор команд, которые передаются с использованием `-X` флага для `send-keys` команды, например, команда режима копирования `start-of-line` перемещает курсор в начало строки и привязывается к `C-a` в `copy-mode` таблице ключей:

```
$ tmux lsk -Tcopy-mode C-a
bind-key -T copy-mode C-a send-keys -X start-of-line
```

Полный список команд режима копирования [доступен на странице руководства](#). Вот подборка:

Команда	<i>emacs(1)</i>	<i>vi(1)</i>	Описание
начать-выбор	C-Space	Космос	Выбор начала
Отмена	q	q	Выйдите из режима копирования
очистить выбор	C-g	Побег	Четкий выбор
копирование по каналу			Скопируйте и передайте в команду, указанную в первом аргументе
копирование-выбор-и-отмена	M-w	Войти	Скопируйте выделенное и выйдите из режима копирования
наведение курсора	Отключено	j	Переместите курсор вниз
курсор-влево	Оставлено	h	Переместите курсор влево
курсор - вправо	Правильно	l	Переместите курсор вправо
наведение курсора вверх	Вверх	k	Переместите курсор вверх
конец строки	C-e	\$	Переместите курсор в конец строки
история-внизу	M->	G	Переместиться в конец истории
история-топ	M-<	g	Перейти к началу истории
промежуточный этап	M-г	M	Перейти к средней строке
следующее слово-конец	M-f	e	Перейти к концу следующего слова
переход по странице вниз	PageDown	C-f	Страница вниз
создание страницы	Страница вверх	C-b	Страница вверх
предыдущее слово	M-b	b	Перейти к предыдущему слову
прямоугольное переключение	R	v	Переключить выделение прямоугольника
поиск-снова	n	n	Повторите последний поиск
поиск в обратном направлении		?	Выполняйте поиск в обратном направлении, первым аргументом является поисковый запрос

Команда	<i>emacs(1)</i>	<i>vi(1)</i>	Описание
поиск-обратный-инкрементный	C-r		Пошаговый поиск в обратном направлении, обычно используется с флагом -i для command-prompt
поиск-вперед		/	При дальнейшем поиске первым аргументом является поисковый запрос
поиск-прямой-инкрементный	C-s		Поиск продвигается постепенно
поиск-обратный	N	N	Повторите предыдущий поиск, но измените направление
запуск линии	C-a	0	Перейти к началу строки

Типы опций

tmux настраивается с помощью параметров. Существует несколько типов параметров:

- Параметры сервера, которые влияют на весь сервер.
- Параметры сеанса, которые влияют на один или все сеансы.
- Параметры окна, которые влияют на одно или все окна.
- Параметры панели, которые влияют на одну или все панели.
- Пользовательские опции, которые не используются tmux, но зарезервированы за пользователем.

Параметры сеанса и окна имеют как глобальный набор параметров, так и набор для каждого сеанса или окна. Если параметр отсутствует в наборе сеанса или окна, используется глобальный параметр. Параметры панели аналогичны, за исключением того, что также отмечены параметры окна.

При настройке tmux чаще всего задаются параметры сервера и глобального сеанса или окна. В этом документе рассматриваются только они.

Отображение параметров

Параметры отображаются с помощью show-options команды. Флаг -g показывает глобальные параметры. Он может отображать параметры сервера, сеанса или окна.:

- -s показывает параметры сервера:

```
$ tmux show -s
backspace C-?
buffer-limit 50
...
```

- -g без других флагов отображаются глобальные параметры сеанса:

```
$ tmux show -g
activity-action other
assume-paste-time 1
...
```

- `-g` и `-w` вместе показывают параметры глобального окна:

```
$ tmux show -wg
aggressive-resize off
allow-rename off
...
```



Значение отдельной опции может быть отображено путем присвоения ее имени `show-option`. Когда указано название опции, указывать `-s` или `-w` не обязательно, потому что `tmux` может определить это по названию опции. Например, чтобы показать `status` опцию:

```
$ tmux show -g status
status on
```



### Изменение параметров

Параметры устанавливаются или отключаются с помощью `set-option` команды. Как и `show-option`, нет необходимости указывать `-s` or `-w`, потому что `tmux` может определить это по названию опции. `-g` необходимо установить глобальные параметры сеанса или окна; для параметров сервера это ничего не делает.

Чтобы установить параметр `status`:

```
set -g status off
```



Или `default-terminal` вариант:

```
set -s default-terminal 'tmux-256color'
```



Флаг `-u` отключает параметр. Отключение глобального параметра восстанавливает его до значения по умолчанию, например:

```
set -gu status
```



### Форматы

Многие параметры используют форматы. Форматы предоставляют мощный синтаксис для настройки отображения текста на основе различных атрибутов сервера `tmux`, сеанса, окна или панели. Форматы заключаются в `#{}` в строковых параметрах или в виде одной заглавной буквы, например `#F`. Это значение по умолчанию `status-right` для нескольких форматов:

```
$ tmux show -s status-right
status-right "#{?window_bigger,[#{window_offset_x}#,#{window_offset_y}] ,}\#{=21:pane_title}\n\nH:M %d-%b-%y"
```



Форматы описаны [в этом документе](#) и [на странице руководства](#).

### Встроенные команды

Некоторые параметры могут содержать встроенные команды командной строки. Это ограничено параметрами строки состояния, такими как `status-left`. Встроенные команды командной строки заключены в `#()`. Они могут либо:

1. Выведите строку и завершите работу, в этом случае строка будет отображаться в строке состояния, и команда будет выполняться с интервалами для ее обновления. Например:

```
set -g status-left '#{uptime}'
```



Максимальный интервал устанавливается параметром `status-interval`, но команды могут также выполняться раньше, если требуется `tmux`. Команды не будут выполняться чаще одного раза в секунду.

2. Продолжайте работать и выводите строку всякий раз, когда это необходимо, например:

```
set -g status-left '#{while ;; do uptime; sleep 1; done}'
```



Обратите внимание, что обычно нет необходимости использовать встроенную команду для даты и времени, поскольку `tmux` сам расширяет форматы даты, такие как `%H` и `%S`, в параметрах строки состояния. Если используется команда типа `date(1)`, любые `%s` должны быть удвоены как `%%`.

## Цвета и стили

`tmux` позволяет настраивать цвет и атрибуты текста с помощью простого синтаксиса, это называется стилем. Стили появляются в двух местах.:

- В настройках, таких как `status-style`.
- Заключенный в `#[]` в значении параметра, это называется встроенным стилем (см. Следующий раздел).

В стиле есть ряд терминов, разделенных пробелами или запятыми, наиболее полезными из которых являются:

- `default` используется цвет по умолчанию; он должен отображаться сам по себе. Цвет по умолчанию часто устанавливается другим параметром, например, для встроенных стилей в `status-left` параметре это `status-style`.
- `bg` задает цвет фона. Цвет также задается, например `bg=red`.
- `fg` задает цвет переднего плана. Как и `bg`, задается цвет: `fg=green`.
- `bright` или `bold`, `underscore`, `reverse`, `italics` установите атрибуты. Они отображаются отдельно, например: `bright,reverse`.

Цвета могут быть одним из `black`, `red`, `green`, `yellow`, `blue`, `magenta`, `cyan`, `white`, `brightred`, `brightyellow`, `colour0` для стандартных цветов терминала; `colour255`, `default` и так далее для ярких вариантов; `#882244 to,,,`, Для цветов из 256-цветовой палитры;,,,,, для цвета по умолчанию; или шестнадцатеричным цветом RGB, таким как,,,,,,.

Остальные термины стиля описаны [на странице руководства](#).

Например, чтобы сделать фон строки состояния синим, используйте опцию `status-style`:

```
set -g status-style 'bg=blue'
```

Встроенные стили

Встроенные стили включены в другой промежуточный параметр `#[ и ]` . Каждый из них изменяет стиль следующего текста до следующего встроенного стиля или конца текста.

Например, чтобы выделить текст красным и синим в `status-left` :

```
set -g status-left 'default #[fg=red] red #[fg=blue] blue'
```

Поскольку это длинный файл, необходимо также увеличить параметр `status-left-length` :

```
set -g status-left-length 100
```

Встроенные стили могут использоваться условно, например, для отображения `P` красным цветом, если был нажат префикс, или в стиле по умолчанию, если нет:

```
set -g status-left '#{?client_prefix,#[bg=red],}P#[default] [{session_name}] '
```

Список полезных опций

Это краткий список наиболее часто используемых параметров `tmux`, помимо параметров стиля:

Опция	Тип	Описание
base-index	сессия	Если задано, то индексы Windows начинаются с этого значения, а не с 0
buffer-limit	сервер	Максимальное количество сохраняемых автоматических буферов по умолчанию равно 50
default-terminal	сервер	Значение по умолчанию для <code>TERM</code> переменной окружения внутри <code>tmux</code>
display-panes-time	окно	Время в миллисекундах, за которое отображаются номера панелей <code>C-b q</code>
display-time	сессия	Время в миллисекундах, в течение которого отображаются сообщения в строке состояния
escape-time	сервер	Время ожидания <code>tmux</code> после получения <code>Escape</code> ключа, чтобы проверить, является ли он частью более длинной последовательности ключей
focus-events	сервер	Отправляются ли <code>tmux</code> последовательности клавиш фокусировки при изменении активной панели и при получении с внешнего терминала, если он их поддерживает
history-limit	сессия	Максимальное количество строк, сохраняемых в истории для каждой панели
mode-keys	окно	Используются ли в режиме копирования привязки клавиш <i>emacs(1)</i> или



Опция	Тип	Описание
		<i>vi(1)</i>
mouse	сессия	Если мышь включена
pane-border-status	окно	Отображается ли строка состояния на границе каждой панели: top или bottom
prefix	сессия	Префиксный ключ по умолчанию равен C-b
remain-on-exit	окно	Автоматически ли отключаются панели при выходе программы, запущенной в
renumber-windows	сессия	Если on , нумерация окон автоматически перенумеровывается, чтобы закрыть все пробелы в списке окон
set-clipboard	сервер	Следует ли tmux пытаться установить внешний буфер обмена X (7) при копировании текста и поддерживает ли это внешний терминал
set-titles	сессия	Если on , tmux установит название внешнего терминала
status	сессия	Видна ли строка состояния, если
status-keys	сессия	Независимо от того, используются ли в командной строке привязки клавиш <i>emacs(1)</i> или <i>vi(1)</i>
status-interval	сессия	Максимальное время в секундах до перерисовки строки состояния
status-position	сессия	Положение строки состояния: top или bottom
synchronize-panes	окно	Если on ввод текста на какой-либо панели в окне отправляется на все панели в окне - следует соблюдать осторожность при выборе этой опции!
terminal-overrides	сервер	Любые возможности tmux должны переопределяться из TERM , предоставленных для внешнего терминала

Список параметров стиля и формата

Это список наиболее часто используемых стилей и форматов tmux:

Опция	Тип	Описание
display-panes-active-colour	сессия	Стиль номера активной панели для C-b q
display-panes-colour	сессия	Стиль номеров панелей, помимо активной панели для C-b q
message-style	сессия	Стиль сообщений, отображаемых в строке состояния и командной строке
mode-style	окно	Стиль выделения в режиме копирования
pane-active-border-style	окно	Стиль границы активной панели

Опция	Тип	Описание
pane-border-format	окно	Установлен формат текста, который отображается в строке состояния границы панели, если pane-border-status
pane-border-style	окно	Стиль границ панели, кроме активной панели
status-left-length	сессия	Максимальная длина оставшейся строки состояния
status-left-style	сессия	Стиль строки состояния оставлен
status-left	сессия	Формат текста в строке состояния оставлен
status-right-length	сессия	Максимальная длина строки состояния справа
status-right-style	сессия	Стиль строки состояния правильный
status-right	сессия	Формат текста в строке состояния правильный
status-style	сессия	Стиль строки состояния в целом и отдельных ее частей может быть переопределен более конкретными параметрами, такими как status-left-style
window-active-style	окно	Стиль цвета по умолчанию для активной панели в окне
window-status-current-format	окно	Формат текущего окна в списке окон
window-status-current-style	окно	Стиль текущего окна в списке окон
window-status-format	окно	Формат окон в списке окон, кроме текущего окна
window-status-separator	окно	Разделитель между окнами в списке окон
window-status-style	окно	Стиль окон в списке окон, кроме текущего окна
window-style	окно	Стиль цвета панелей в окне по умолчанию, за исключением активной панели

### Общие изменения конфигурации

В этом разделе показаны примеры некоторых распространенных изменений конфигурации для `.tmux.conf`.

#### Изменение префиксного ключа

Префиксный ключ устанавливается с помощью опции `prefix`. С-b Клавиша также привязана к `send-prefix` команде в таблице префиксных клавиш, поэтому, дважды нажав С-b, она переходит на активную панель. Чтобы изменить на С-a:

```
set -g prefix C-a
unbind C-b
bind C-a send-prefix
```



## Настройка строки состояния

Существует множество вариантов настройки строки состояния. Самые простые варианты:

- Отключите строку состояния: `set -g status off`
- Переместите это в начало: `set -g status-position top`
- Установите цвет фона на красный: `set -g status-style bg=red`
- Измените текст справа только на время: `set -g status-right '%N:%M'`
- Подчеркните текущее окно: `set -g window-status-current-style 'underscore'`

## Настройка границы панели

Цвета границ панели могут быть заданы:

```
set -g pane-border-style fg=red
set -g pane-active-border-style 'fg=red,bg=yellow'
```



В каждой области может быть придан статус соответствии с `pane-border-status` вариантом, для например, чтобы отобразить панель заголовка жирным шрифтом:

```
set -g pane-border-status top
set -g pane-border-format '[bold]#{pane_title}#[default]'
```



## *vi* (1) привязки клавиш

tmux поддерживает привязки клавиш на основе *vi* (1) для режима копирования и командной строки. Есть два варианта, которые устанавливают привязки клавиш:

1. `mode-keys` устанавливает привязки клавиш для режима копирования. Если это значение равно `vi`, то в режиме копирования используется `copy-mode-vi` таблица ключей; в противном случае используется `copy-mode` таблица ключей.
2. `status-keys` устанавливает привязки клавиш для командной строки.

Если для любой из переменных окружения `VISUAL` или `EDITOR` установлено значение, содержащее что-либо *vi* (например, `vi`, `vim`, `nvi`) при первом запуске сервера `tmux`, для обоих этих параметров установлено значение `vi`.

Чтобы настроить оба режима на использование *vi* (1) клавиш:

```
set -g mode-keys vi
set -g status-keys vi
```



## Поведение при копировании мышью

При перетаскивании мыши для копирования текста `tmux` копирует текст и выходит из режима копирования при отпускании кнопки мыши. Альтернативные варианты поведения настраиваются путем изменения `MouseDragEnd1Pane` привязки клавиш. Три наиболее полезных:

1. Не копируйте и не очищайте выделение и не выходите из режима копирования, когда отпущена мышь. Для копирования выделения необходимо использовать клавиатуру:

```
unbind -Tcopy-mode MouseDragEnd1Pane
```



2. Скопируйте и снимите выделение, но не выходите из режима копирования:

```
bind -Tcopy-mode MouseDragEnd1Pane send -X copy-selection
```



3. Скопируйте, но не снимайте выделение:

```
bind -Tcopy-mode MouseDragEnd1Pane send -X copy-selection-no-clear
```



## Другие функции

`tmux` обладает большим набором функций и команд, не упомянутых в этом документе, многие из которых позволяют создавать мощные сценарии. Вот список некоторых, с которыми, возможно, стоит ознакомиться подробнее.:

- Оповещения: `monitor-activity` , `monitor-bell` , `monitor-silence` `activity-action` , `bell-action` и другие опции.
- Параметры для отдельного сеанса, окон и панелей.
- Перемещение панелей с помощью `join-pane` и `break-pane` .
- Отправка ключей к панелям с помощью `send-keys` .
- Опция командной строки `history-file` .
- Сохраненные строки макета с помощью `select-layout` .
- Последовательности команд (разделенные символом `;`): `select-window`; `kill-window` .
- Синтаксис файла конфигурации: `{}` , `%if` и так далее.
- Привязки клавиш мыши: `MouseDown1Pane` и так далее.
- Блокировка: `lock-command` , `lock-after-time` и другие опции.
- Захват содержимого панели с помощью `capture-pane` и конвейеризация с помощью `pipe-pane` .
- Связывание Windows: команда `link-window` .
- Группы сеансов: `-t` флаг для `new-session` .
- Обновление окна и панелей с помощью `respawn-window` и `respawn-pane` .

- Пользовательские меню с `display-menu` командой и пользовательские подсказки с `command-prompt` и `confirm-before` .
- Разные ключевые таблицы: `bind-key` и `-T` флаг для `switch-client` .
- Пустые панели: `split-window` с пустой командой и `-I to display-message` .
- Хуки: `set-hook` и `show-hooks` .
- Синхронизация скриптов с `wait-for` .

▼ Pages 11
<div>Find a page...</div>
▸ <a href="#">Home</a>
▸ <a href="#">Advanced Use</a>
▸ <a href="#">Clipboard</a>
▸ <a href="#">Contributing</a>
▸ <a href="#">Control Mode</a>
▸ <a href="#">FAQ</a>
▸ <a href="#">Formats</a>
▼ <a href="#">Getting Started</a> <div>Getting started</div> <div>About tmux</div> <div>About this document</div> <div>Other documentation and help</div> <div>Basic concepts</div> <div>The tmux server and clients</div> <div>Sessions, windows and panes</div> <div>Summary of terms</div> <div>Using tmux interactively</div> <div>Creating sessions</div> <div>The status line</div> <div>The prefix key</div> <div>Help keys</div> <div>Commands and flags</div> <div>The command prompt</div> <div>Attaching and detaching</div> <div>Listing sessions</div> <div>Killing tmux entirely</div> <div>Creating new windows</div> <div>Splitting the window</div> <div>Changing the current window</div> <div>Changing the active pane</div> <div>Choosing sessions, windows and panes</div>

- Detaching other clients
- Killing a session, window or pane
- Renaming sessions and windows
- Swapping and moving
- Resizing and zooming panes
- Window layouts
- Copy and paste
- Finding windows and panes
- Using the mouse
- Configuring tmux
  - The configuration file
  - Key bindings
  - Copy mode key bindings
  - Types of option
  - Showing options
  - Changing options
  - Formats
  - Embedded commands
  - Colours and styles
  - Embedded styles
  - List of useful options
  - List of style and format options
- Common configuration changes
  - Changing the prefix key
  - Customizing the status line
  - Configuring the pane border
  - vi(1) key bindings
  - Mouse copying behaviour
- Other features

▸ <a href="#">Installing</a>
▸ <a href="#">Ключи-модификаторы</a>
▸ <a href="#">Рецепты</a>

Клонируйте эту вики локально

https://github.com/tmux/tmux/wiki.git