A beginner's guide to tmux

redhat.com/sysadmin/introduction-tmux-linux

September 13, 2022

Make your Linux terminal more useful with tmux, a terminal multiplexer that allows you to run multiple Linux programs over a single connection.

Posted: September 13, 2022 |

undefined

| Ricardo Gerardi (Editorial Team, Sudoer alumni, Red Hat)

Image

Photo by Sora Shimazaki from Pexels

Tmux is a terminal multiplexer; it allows you to create several "pseudo terminals" from a single terminal. This is very useful for running multiple programs with a single connection, such as when you're remotely connecting to a machine using Secure Shell (SSH).

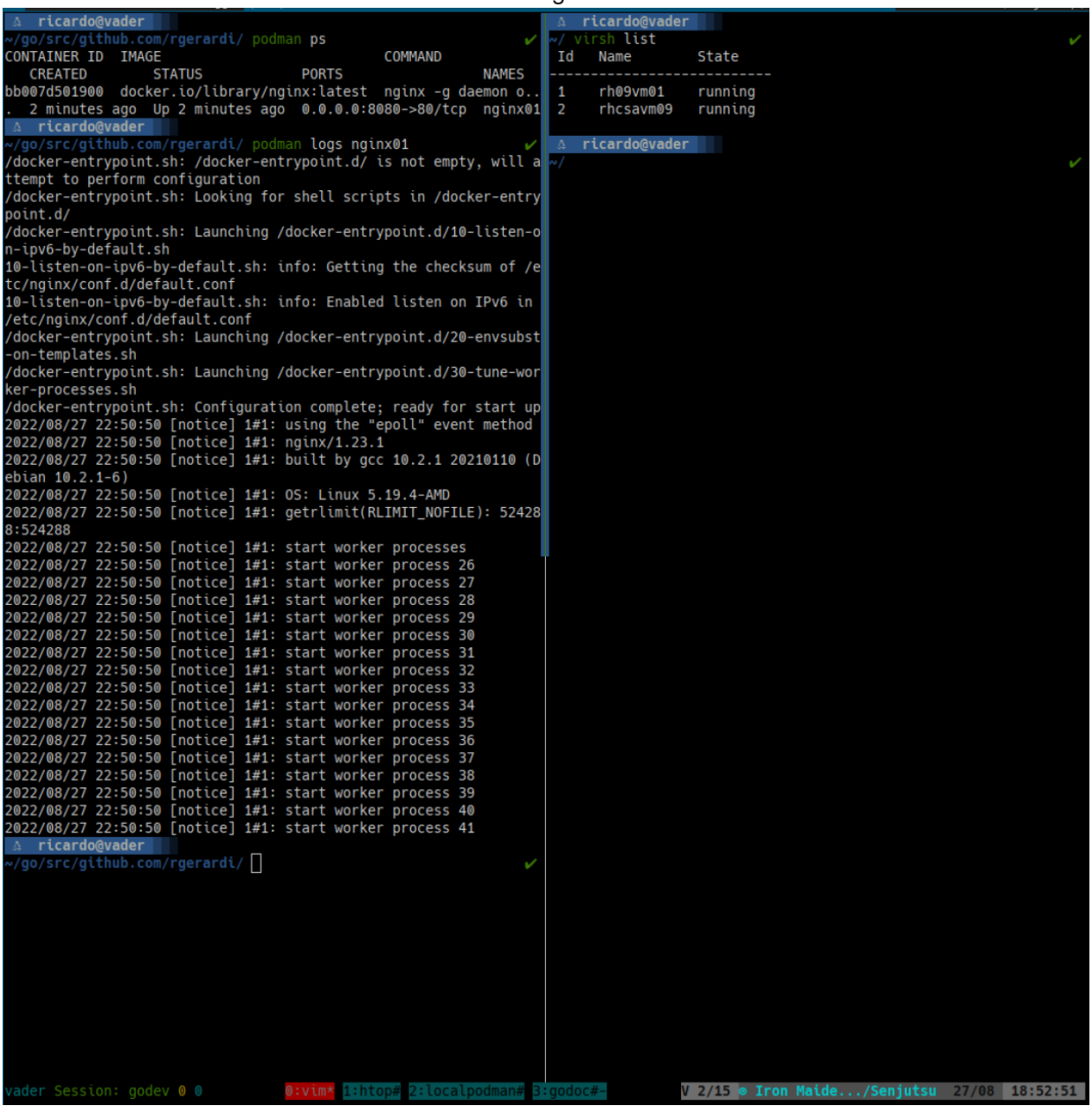
Tmux also decouples your programs from the main terminal, protecting them from accidentally disconnecting. You can detach tmux from the current terminal, and all your programs will continue to run safely in the background. Later, you can reattach tmux to the same or a different terminal.

In addition to its benefits with remote connections, tmux's speed and flexibility make it a fantastic tool to manage multiple terminals on your local machine, similar to a window manager. I've been using tmux on my laptops for over eight years. Some of tmux's features that help me and increase my productivity include:

- Fully customizable status bar
- Multiple window management
- Splitting window in several panes
- Automatic layouts
- Panel synchronization
- Scriptability, which allows me to create custom tmux sessions for different purposes

Here's an example of a customized tmux session:

Image



(Ricardo Gerardi, CC BY-SA 4.0)

Tmux offers some of the same functionality found in Screen, which has been deprecated in some Linux distributions. Tmux has a more modern code base than Screen and offers additional customization capabilities.

Now that you know some of tmux's benefits, I'll show you how to install and use it.

Install tmux

Tmux is available in the standard repositories with Fedora and Red Hat Enterprise Linux (RHEL), starting with RHEL 8. You can install it using DNF:

```
$ sudo dnf -y install tmux
```

It's also available with many other Linux distributions, and you should be able to install it by using your favorite distribution package manager. For other operating systems, consult the tmux installation guide.

[Download now: A sysadmin's guide to Bash scripting.]

Get started with tmux

To start using tmux, type `tmux` on your terminal. This command launches a tmux server, creates a default session (number 0) with a single window, and attaches to it.

```
$ tmux
```

Image

(Ricardo Gerardi, CC BY-SA 4.0)

Now that you're connected to tmux, you can run any commands or programs as you normally would. For example, to simulate a long-running process:

```
$ c=1

$ while true; do echo "Hello $c"; let c=c+1; sleep 1; done
Hello 1
Hello 2
Hello 3
```

You can detach from your tmux session by pressing **Ctrl+B** then **D**. Tmux operates using a series of keybindings (keyboard shortcuts) triggered by pressing the "prefix" combination. By default, the prefix is **Ctrl+B**. After that, press **D** to detach from the current session.

```
[detached (from session 0)]
```

You're no longer attached to the session, but your long-running command executes safely in the background. You can list active tmux sessions with `tmux ls`:

```
$ tmux ls

0: 1 windows (created Sat Aug 27 20:54:58 2022)
```

[Learn how to manage your Linux environment for success.]

You can disconnect your SSH connection at this point, and the command will continue to run. When you're ready, reconnect to the server and reattach to the existing tmux session to resume where you left off:

```
$ tmux attach -t 0
Hello 72
Hello 73
Hello 74
Hello 75
Hello 76
^C
```


As you can see, the command continued to run and print messages on the screen. You can type **Ctrl+C** to cancel it.

All tmux commands can also be abbreviated, so, for example, you can enter `tmux a` , and it will work the same as `tmux attach`.

This functionality alone makes tmux a great tool, but it has even more to offer, including its default keybindings.

Basic tmux keybindings

Tmux provides several keybindings to execute commands quickly in a tmux session. Here are some of the most useful ones.

First, create a new tmux session if you're not already in one. You can name your session by passing the parameter `-s {name}` to the `tmux new` command when creating a new session:

```
$ tmux new -s Session1
```

- **Ctrl+B D** — Detach from the current session.
- **Ctrl+B %** — Split the window into two panes horizontally.
- **Ctrl+B "** — Split the window into two panes vertically.
- **Ctrl+B Arrow Key** (Left, Right, Up, Down) — Move between panes.
- **Ctrl+B X** — Close pane.
- **Ctrl+B C** — Create a new window.
- **Ctrl+B N** or **P** — Move to the next or previous window.
- **Ctrl+B 0 (1,2...)** — Move to a specific window by number.
- **Ctrl+B :** — Enter the command line to type commands. Tab completion is available.
- **Ctrl+B ?** — View all keybindings. Press **Q** to exit.
- **Ctrl+B W** — Open a panel to navigate across windows in multiple sessions.

For additional keybindings, consult the tmux man pages.

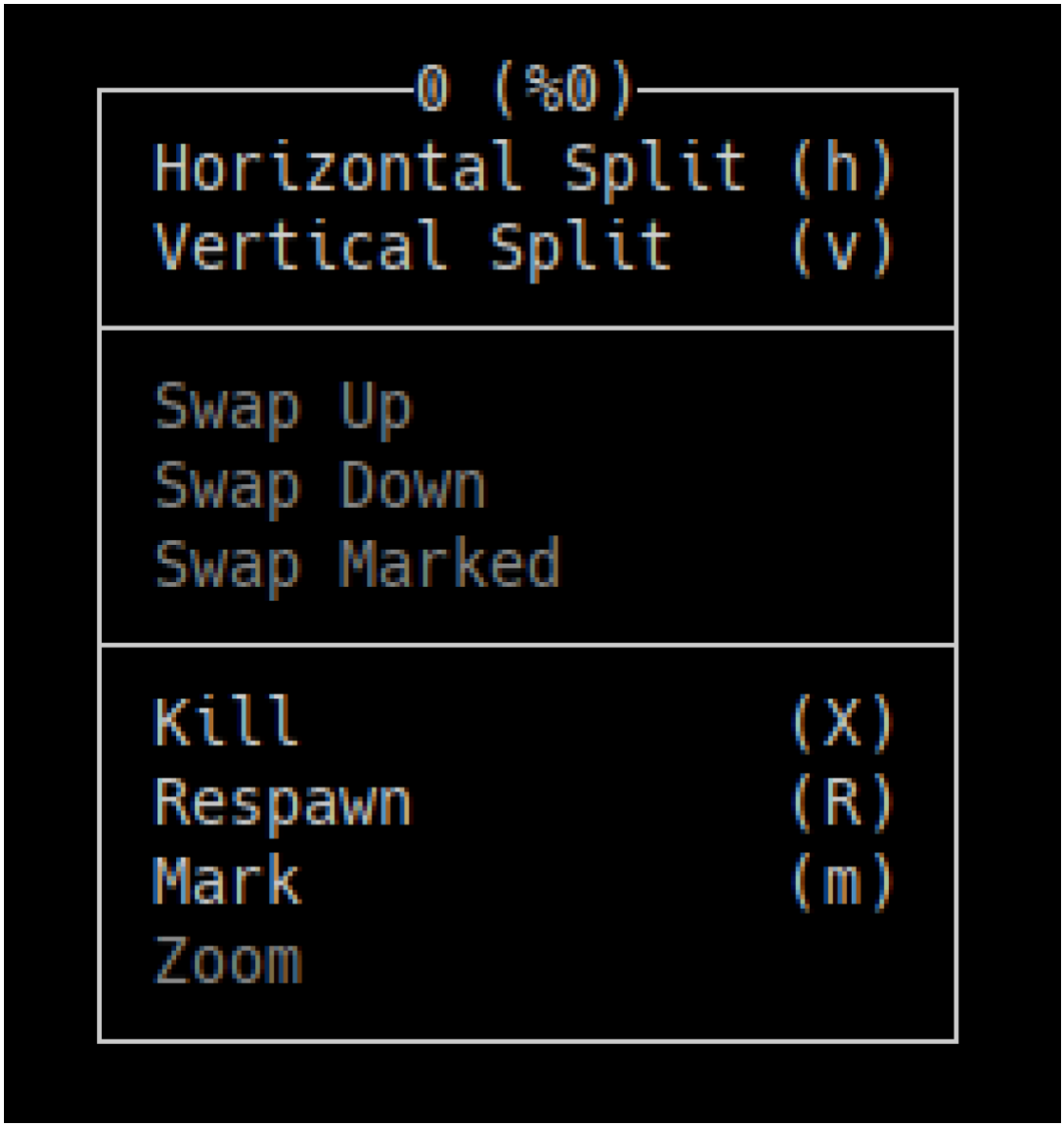
[Download the tmux cheat sheet to keep the keybindings at your fingertips.]

Use the mouse

Tmux is most often used with the keyboard, and it provides many keybindings to make it easier to execute commands, create new panes, and resize them. If you prefer using the mouse, tmux also allows that, although the mouse is disabled by default. To enable it, first enter command mode by typing **Ctrl+B :**, then toggle the mouse on (or off) with the command `set -g mouse`.

Now you can use the mouse to switch between panes and windows and resize them. Starting with tmux version 3, you can also right-click with the mouse and open a context menu:

Image



This menu changes according to what's on the screen under the mouse cursor when clicked.

[Keep your most commonly used commands handy with the Linux commands cheat sheet.]

Configure tmux

You can change the tmux configuration permanently by modifying the tmux configuration file. By default, this file is located at `$HOME/.tmux.conf`.

For example, the default prefix key combination is **Ctrl+B**, but sometimes this combination is a little awkward to press, and it requires both hands. You can change it to something different by editing the configuration file. I like to set the prefix key to **Ctrl+A**. To do this, create a new configuration file and add these lines to it:

```
$ vi $HOME/.tmux.conf

# Set the prefix to Ctrl+a
set -g prefix C-a

# Remove the old prefix
unbind C-b

# Send Ctrl+a to applications by pressing it twice
bind C-a send-prefix

:wq
```

When you start a tmux session on this machine, you can execute the commands listed above by pressing **Ctrl+A** first. Use the configuration file to change or add other tmux keybindings and commands.

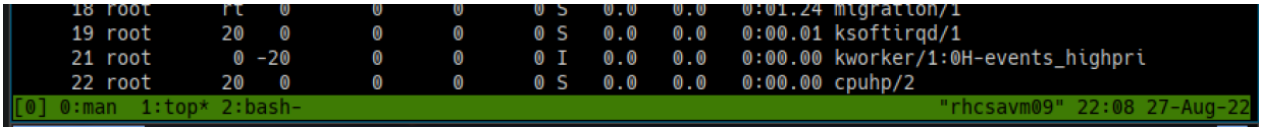
[Get the guide to installing applications on Linux.]

Customize the status bar

Tmux's status bar is fully customizable. You can change the colors of each section and what is displayed. There are so many options that it would require another article to cover them, so I'll start with the basics.

The standard green color for the entire status bar makes it difficult to see the different sections. It's particularly difficult to see how many windows you have open and which one is active.

Image



(Ricardo Gerardi, CC BY-SA 4.0)

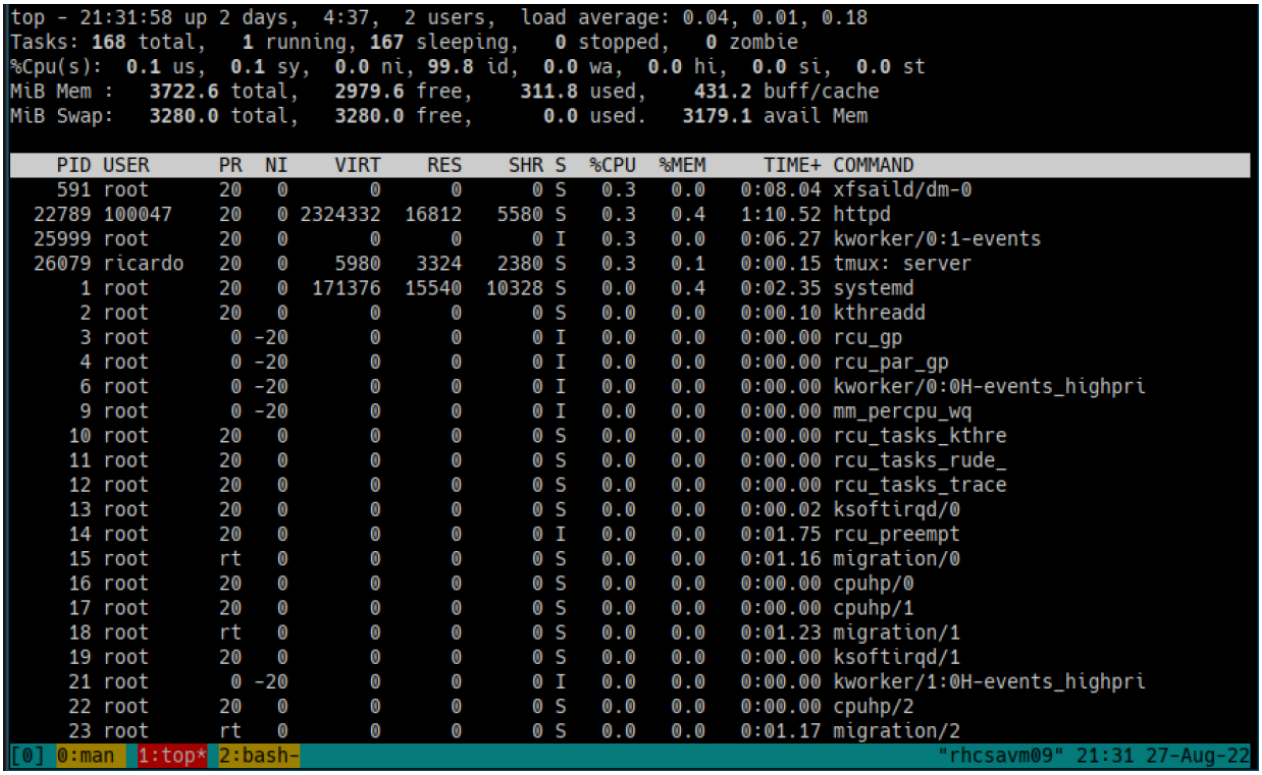
You can change that by updating the status bar colors. First, enter command mode by typing **Ctrl+B :** (or **Ctrl+A :** if you made the prefix configuration change above). Then change the colors with these commands:

- Change the status bar background color: `set -g status-bg cyan`
- Change inactive window color: `set -g window-status-style bg=yellow`
- Change active window color: `set -g window-status-current-style bg=red,fg=white`

Add these commands to your configuration file for permanent changes.

With this configuration in place, your status bar looks nicer, and it's much easier to see which window is active:

Image



(Ricardo Gerardi, CC BY-SA 4.0)

What's next

Tmux is a fantastic tool to safeguard your remote connections and is useful when you spend a long time using the terminal. This article covers only the basic functionality, and there is much more to explore. For additional information about tmux, consult its official wiki page.

You can also expand tmux's functionality with extra-official plugins. These plugins add more commands, integrate with applications such as Vim, and add new functionality to the status bar. For more information, consult the tmux plugins project.

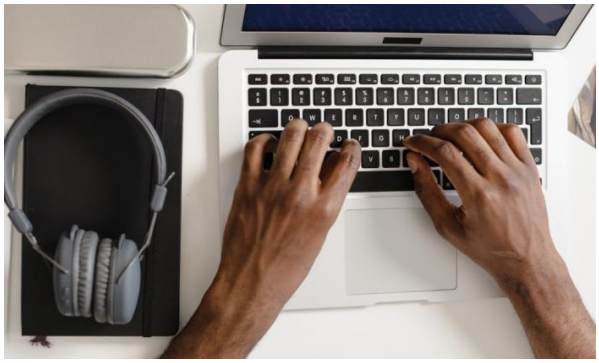
Image

Posted: August 15, 2022

Author: Seth Kenlon (Editorial Team, Red Hat)

Image

How to get started with the Vi editor



Once you've committed Vi's keyboard shortcuts to muscle memory, watch how fast you work.

Posted: January 25, 2022

Author: Seth Kenlon (Editorial Team, Red Hat)

Image

Linux tool alternatives: 6 replacements for traditional favorites

Consider swapping Linux tools for these alternatives that provide more features and functionality.

Posted: July 27, 2022

Author: Jose Vicente Nunez (Sudoer)

Topics: **Command line utilities** **Linux**

Ricardo Gerardi

Ricardo Gerardi is Technical Community Advocate for Enable Sysadmin and Enable Architect. He was previously a senior consultant at Red Hat Canada, where he specialized in IT automation with Ansible and OpenShift. More about me