

# Installing and using tmux in Slackware

*tmux* is a terminal multiplexer. It enables a number of terminals (or windows) to be accessed and controlled from a single terminal. It is intended to be a simple, modern, BSD-licensed alternative to programs such as GNU Screen.

*tmux* uses a client-server model. The server holds multiple sessions and each window is an independent entity which may be freely linked to multiple sessions, moved between sessions and otherwise manipulated. Each session may be attached to (display and accept keyboard input from) multiple clients. Major features include:

- A powerful, consistent, well-documented and easily scriptable command interface.
- A window may be split horizontally and vertically into panes.
- Panes can be freely moved and resized, or arranged into preset layouts.
- Support for UTF-8 and 256-colour terminals.
- Copy and paste with multiple buffers.
- Interactive menus to select windows, sessions or clients.
- Change the current window by searching for text in the target.
- Terminal locking, manually or after a timeout.
- A clean, easily extended, BSD-licensed codebase, under active development.

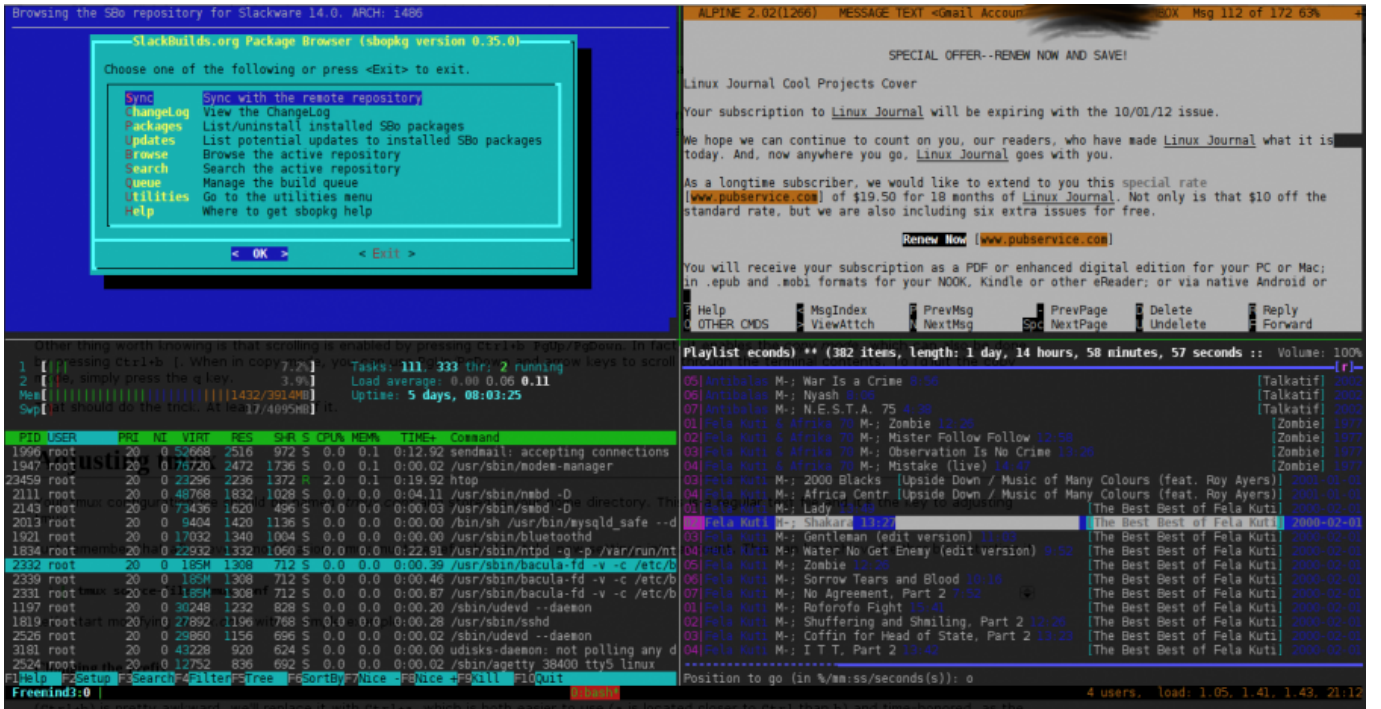
## Installation

A ***tmux*** SlackBuild is available at your favorite SlackBuilds [repository](#). The installation is pretty straightforward. First, make sure all the dependencies are installed (at the time of this writing, ***libevent*** is the only dependency). Download the SlackBuild archive for *tmux*, extract it in your build environment. Next, download the source of *tmux* from their [website](#) and place it in the extracted directory. Run the script `./tmux.SlackBuild` and finally use ***installpkg*** to install the finished package that should be located in `/tmp`. You can consider ***sbopkg*** as a helper tool for installing the SlackBuild.

## Configuration

You can configure ***tmux*** through a user specific configuration file, `~/.tmux.conf`, or a system configuration file, `/etc/tmux.conf`. Sample configuration files are available in `/usr/share/doc/tmux-X.X/examples/`, where `X.X` depends on the version of ***tmux*** you installed. You might find just what you need in there.

*Tmux* is very flexible and the ability to customize its behavior through scripting is pretty neat. This is a *tmux* screenshot showing four panes:



The **tmux** man page is pretty extensive and has very good documentation for those of us enjoying heavy customization. If you have used **GNU Screen** before, you must be familiar with the concept of prefix key: the same concept applies to **tmux**. It is a combination of keys you use to control **tmux** from an attached client. In **GNU Screen**, the default prefix key is **Ctrl+A** while **tmux** has **Ctrl+B** as default prefix. So, for users with long term **GNU Screen** addiction, you can redefine **tmux** prefix by adding the command below to your configuration file `~/tmux.conf`:

```
set -g prefix Ctrl-a
```

Few **tmux** key bindings:

- **Ctrl+B** **C** Create a new window.
- **Ctrl+B** **D** Detach the current client.
- **Ctrl+B** **&** Kill the current window.
- **Ctrl+B** **%** Split the current window vertically into two panes.
- **Ctrl+B** **|** Split the current window horizontally into two panes.
- **Ctrl+B** **O** Select the next pane in the current window.
- **Ctrl+B** **{** Swap the current pane with the previous pane.
- **Ctrl+B** **}** Swap the current pane with the next pane.
- **Ctrl+B** **?** List all key bindings.

The **tmux** man page contains more detailed information about **tmux** usage and customization. It has every thing to get you started.

## Sources

- [tmux FAQ](#)
- [tmux website](#)
- [A tmux tutorial \(Part 1\)](#)

- [Another tmux tutorial](#)
- [Yet another tmux tutorial](#)
- Originally written by [escaflow](#)

[howtos](#), [software](#), [tmux](#), [author escaflow](#)

From:

<https://docs.slackware.com/> - **SlackDocs**

Permanent link:

<https://docs.slackware.com/howtos:software:tmux>

Last update: **2012/12/28 05:28 (UTC)**

