

APPS@UCU

Linux course

Tools overview

Morhunenko Mykola



Contents

- 1 Vim
- 2 Tmux
- 3 Ranger
- 4 i3
- 5 Package managers
- 6 Sources

- In this presentation, we will overview some tools that are available on all Linux distributions
- All of them have a high barrier to entry as Linux itself, but when you are there - you won't imagine your life without that tools
- Example: it's not a one-day task to learn how to move around your system, but after few months of practice working with CLI, GUI for you will be as slow as a turtle is slow in comparison with a rabbit



- So let's start with the text editor
- You have heard about **vim** , haven't you?
- But let's start with **vi**
- Vi is a part of POSIX
- It's totally CLI editor (forget that you have a mouse)
- There are shortcuts for everything
- If there is no, you can create them for yourself
- Every good enough 21'st century editor has an extension for a **vi mode**
- But almost nobody uses it - only on some low memory and low power machines. So we move to vim



- **Vim** stands for 'Vi IMproved'
- According to Linux Journal survey, 38% (in average for 2009-2018) of respondents vote for vim as the best editor
- It has much more features, than **vi**, including more commands, scriptable syntax highlighting and extensions, graphical interface (and a mouse support, but don't use it)
- As **vi**, it has six modes - normal, visual, insert, command-line, select, and ex (yes, not only NORMAL and INSERT)
- Because of a huge community (38% of world's best geeks) **vim** became a powerful IDE with thousands of extensions (syntax highlight, autocompletion, spell checking, project tree etc)
- The most powerful tool of **vim** is inside - its shortcuts. You can make your work dozens of times faster without a touchpad and a mouse

- Neovim is just a fork of Vim with some Python extensions
- And cool logo =)
- Also Neovim is a community-driven text editor, while Vim is a project of only one person - [Bram Moolenaar](#)
- One 'expert' on reddit wrote that:
"Neovim exists to convince Bram to push new features to Vim"
And I mostly agree with him.

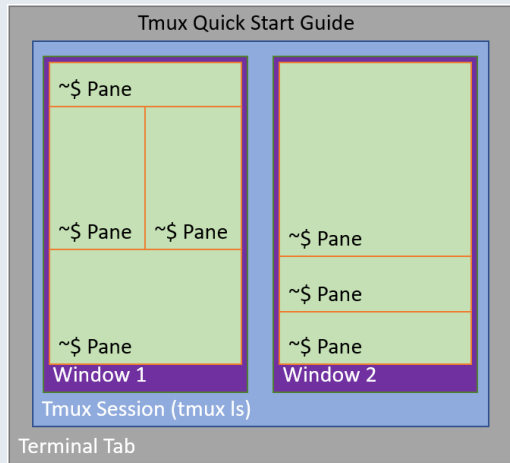


Tmux

- **TMUX** stands for Terminal MULTipleXer
- There are some other (screen, Konsole, etc.), but they are not so good as TMUX
- Why do we need it?
- As you continue your practice in CLI, you can notice that it's not enough to have only one terminal window
- With this much multitasking going on, we want to have more terminals. So people create a **terminal multiplexor**
- What TMUX can do?
 - Not only split and stack tab but also make tabs
 - Continue running programs in the background
 - With extensions you can write layout files in **.yaml** format
 - Search through terminal output and move around with Vim shortcuts
 - Other interesting stuff

Tmux

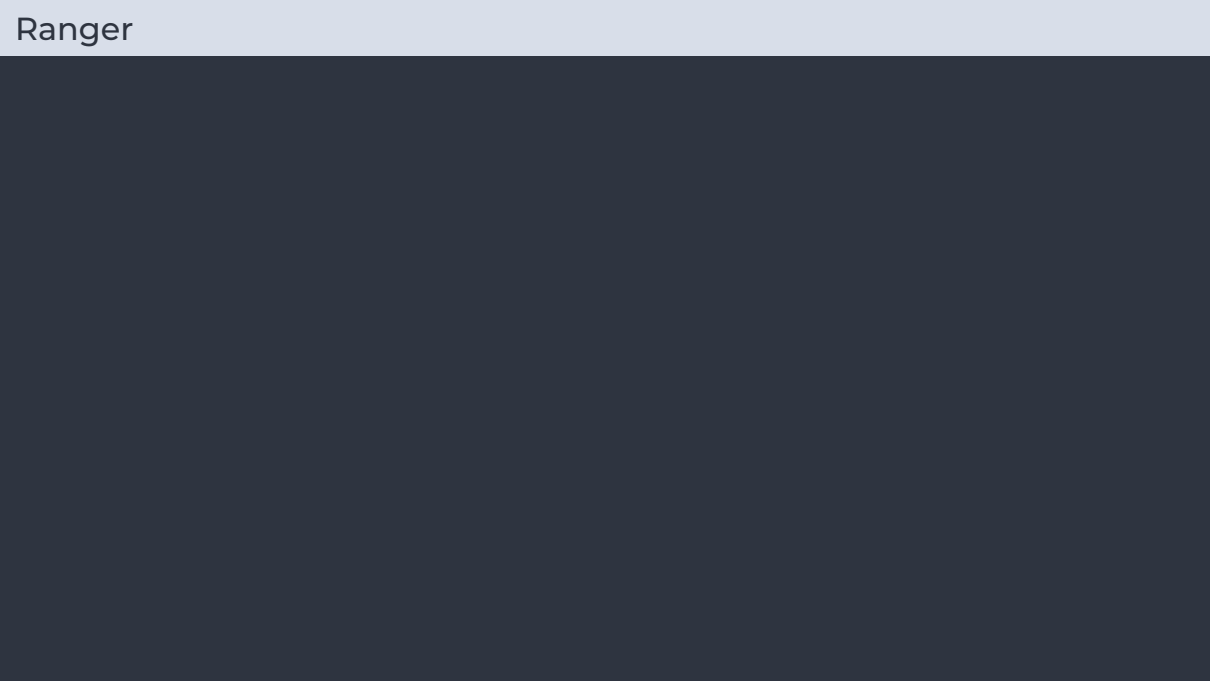
- A Tmux Session with two tmux tabs with multiple tmux panes within each
- As `vim`, tmux has modes - view and command (Ctrl+b be default)
- Every pane has three modes - view, choose and copy
- To enter a `copy mode` - `Ctrl+b [`
- It allows you to use vim keys for moving around and copying text
- For more information see `man tmux` or `linux man page`



For the very beginning:

[tmux sessions]		linuxacademy.local		[tmux windows]		linuxacademy.local	
_ new sessions tmux tmux new tmux new-session tmux new -s sessionname		_ remove sessions tmux kill-ses tmux kill-session -t sessionname		_ windows are like tabs in a browser. Windows exist in sessions and occupy the space of a session screen.		Ctrl + B 0..9 select window by number	
_ attach sessions tmux a tmux att tmux attach tmux attach-session tmux a -t sessionname		_ key bindings Ctrl + B \$ rename session Ctrl + B D detach session Ctrl + B) next session Ctrl + B (previous session		_ key bindings Ctrl + B C create window Ctrl + B N move to next window Ctrl + B P move to previous window Ctrl + B L move to window last used		Ctrl + B ' select window by name Ctrl + B . change window number Ctrl + B , rename window Ctrl + B F search windows Ctrl + B & kill window	
[tmux panes]		linuxacademy.local		[tmux copy mode]		linuxacademy.com	
_ panes are sections of windows that have been split into different screens - just like the panes of a real window!		Ctrl + B ↑ move up to pane Ctrl + B ↓ move down to pane Ctrl + B O go to next pane Ctrl + B ; go to last active pane		_ key bindings Ctrl + B [enter copy mode Ctrl + B] paste from buffer		G go to bottom h move cursor left j move cursor down k move cursor up l move cursor right / search # list paste buffers q quit	
_ key bindings Ctrl + B % vertical split Ctrl + B " horizontal split Ctrl + B → move to pane to the right Ctrl + B ← move to pane to the left		Ctrl + B } move pane right Ctrl + B { move pane left Ctrl + B ! convert pane to window Ctrl + B X kill pane		_ copy mode commands space start selection enter copy selection Esc clear selection g go to top			





Ranger

i3wm





Package managers



Package managers

Sources

Sources

- [Linux journal](#)
- [Termianl Multiplexers](#)
- [Tmux tutorial](#)
- [Tmux Linux man page](#)