



Linux Users Group at NC State University

Fuzzy Finders Lazy Search

Bennett Petzold

LUG @ NC State

November 23, 2021

- What
 - Finds approximate matches (close-enough searches)
 - (Usually) interactive
 - (Sometimes) displays extra info for items
- Why
 - Imperfect memory / laziness
 - Avoiding repeated precise searches
 - Skimming
- Where
 - Command line, editors, IDEs, search engines, etc
- How
 - Rank the items closest to input
 - Different scoring algorithms
 - Usually remove lowest scores

Fuzzy Finders

- Command-Line (fz)

- Command-Line (not-fz)

- Vim Plugin Only

- fzf [1]
 - The one I use (and will discuss)
 - Written in Go
 - Most features
 - Partial input from pipe
 - Vim plugin
 - Emacs plugin (fzf.el)
 - Active development
- fzy [2]
 - Claims faster speeds than fzf
 - Written in C
 - Only complete input
 - Can be integrated with Vim
 - Last commit in October 2020

- Selecta [3]
 - Written in Ruby
 - Can be integrated with Vim
 - Semi-active development
- Skim [4]
 - Written in Rust
 - Vim Plugin (lotabout/skim)
 - Also a library
 - Semi-active development

- Command-T [5]
 - Claims to be the fastest fuzzy finder
 - Written in Ruby
 - Semi-active development
- ctrlp.vim [6]
 - Written in *Vim Script*
 - Semi-active development

fzf Command Line [1]
Calling & Navigating
Input
Flags

```
fzf
cmd **<TAB>
fzf < input
cmd | fzf | other_cmd
export FZF_DEFAULT_OPTS='--option1 arg --option2' # etc
export FZF_DEFAULT_COMMAND='rg --files' # ripgrep backend
```

- CTRL-K / CTRL-P / UP to go up
- CTRL-J / CTRL-N / DOWN to go down
- TAB and SHIFT-TAB to mark multiple items
- ENTER to exit

- No prefix or suffix = fuzzy find
- ' (apostrophe) = exact match
- ^ (caret) = prefix with exact match
- \$ (dollar) = suffix with exact match
- ! (exclamation) = invert exact match
- | (vertical bar) = or
- Prefixes and suffixes separated by spaces
- example: `!^ncsu .tex$ template`
 - “`!^ncsu`” = not starting with “ncsu”
 - “`.tex$`” = ending with “.tex”
 - “`template`” = fuzzy find for “template”

Flags

(fzf Command Line [1])

- -m : Multi select
- -e : Exact, ' (apostrophe) enables fuzzy instead
- -disabled : Don't search
- -algo=v1 : Faster, less optimal algorithm
- -bind : custom key bindings
- -reverse : flipped, with search bar on top
- -layout=reverse-list : flipped, search bar on bottom
- -ansi : process ANSI color codes
- -color= : set colors
- -preview=cmd : show cmd run on cursor line in preview
- -height= : number of lines or percent

Fuzzy Finders in Editors

Options

Using fzf.vim [9]

Options

(Fuzzy Finders in Editors)

- Emacs: `affe.el` [7]
 - Written in Emacs Lisp
- Emacs: `fzf.el` [8]
 - `fzf` frontend
- Vim: `fzf` [9]
 - `fzf` frontend
- Vim: `fzf.vim` [9]
 - Built on top of `fzf`, Vim Script
- Neovim: `telescope.nvim` [10]
 - Written in Lua
 - Choice of `fzf` and `fzy` backend
 - More customizable

Using fzf.vim [9]

(Fuzzy Finders in Editors)

- Need to install “junegunn/fzf” and “junegunn/fzf.vim”

```
Plug 'junegunn/fzf', { 'do': { -> fzf#install() } }  
Plug 'junegunn/fzf.vim'
```

- Useful commands

- :Lines
 - :BLines
 - :Rg / :Ag
 - :Buffers
 - :Files
 - :GFiles
 - :GFiles?
 - :Helptags
 - :Snippets
- Happy ripgrep backend = global .gitignore and/or .ignore

```
export RIPGREP_CONFIG_PATH="$HOME/.config/ripgrep/ripgreprc"  
echo "--ignore-file=$HOME/.gitignore" > $HOME/.config/ripgrep/ripgreprc  
echo "--ignore-file=$HOME/.ignore" > $HOME/.config/ripgrep/ripgreprc
```

Examples
Speeds
Command Line

```
echo line > file
while [ $(bc <<< "${(wc -c < file) < 1024^2 * 100}") -eq 1 ]; do
    cat file > altfile; cat altfile >> file;
done
du -sh file # Should be 160 M
fzf < file # ~5.8 seconds
fzy < file # ~2.5 seconds
sk < file # ~9.5 seconds

echo line > file
while [ $(bc <<< "${(wc -c < file) < 1024^2}") -eq 1 ]; do
    cat file > altfile; cat altfile >> file;
done
du -sh file # Should be 1.3 M
fzf < file # Faster than me
fzy < file # Faster than me
sk < file # Faster than me
```

- fzy is faster for large inputs
- fzf and sk pull up interface immediately, fzy waits

Getting LUG presentations

```
alias crwl="crawl -robots respect -silent"  
crwl https://lug.ncsu.edu/presentations/ | fzf -m | parallel wget
```

Adding ssh certs

```
ssh-add $(ls ~/.ssh/ | fzf -m)
```

Where is that

```
cd $(fzf)  
cd ** //Followed by <TAB>  
$EDITOR ** //Followed by <TAB>  
kill -9 //Followed by <TAB>  
ssh ** //Followed by <TAB>
```


Resources

- [1] fzf
<https://github.com/junegunn/fzf>
- [2] fzy
<https://github.com/jhawthorn/fzy>
- [3] selecta
<https://github.com/garybernhardt/selecta>
- [4] skim
<https://github.com/lotabout/skim>
- [5] Command-T
<https://github.com/wincent/Command-T>
- [6] ctrlp.vim
<https://github.com/ctrlpvim/ctrlp.vim>
- [7] affe.el
<https://github.com/minad/affe>
- [8] fzf.el
<https://github.com/bling/fzf.el>
- [9] fzf.vim
<https://github.com/junegunn/fzf.vim>
- [10] telescope.nvim
<https://github.com/nvim-telescope/telescope.nvim>