Fzf

Christian Külker

2023-04-12

Contents

`fzf` is an interactive filter program that can be used to search for items in a list. For example, it can be used to interactively match filenames in a list of files.

1. Interactive interface: `fzf` provides an easy way to search through large collections of files, command histories, and other types of data.
2. Customizable: `fzf` is customizable via environment variables `FZF_DEFAULT_COMMAND` and `FZF_DEFAULT_OPTS`, allowing users to configure it to suit their specific needs. It has a wide range of options that can be used to add functionality and integrate with other tools.
3. Some external programs, like `zsh`, `vim` and `tmux` have plugins to interact with `fzf`.

  - `fzf-tab`: Replace zsh's default completion selection menu with `fzf`!
  - `fzf-tmux`: https://github.com/junegunn/fzf/tree/master/bin
  - `fzf.vim`: https://github.com/junegunn/fzf.vim

4. Versatile: fzf can be used for a variety of tasks, such as navigating files, searching command history, selecting processes to kill, and more. It can also be integrated with other tools such as vim, tmux, and zsh.

1 Installation

```
aptitude install fzf
```

2 Source

Source: https://github.com/junegunn/fzf

3 Usage

3.1 Safe Results in a File

Search for 'book' in PDF files interactively and save the result in `results.txt`:

```
# Run
find / -name "*.pdf*"|fzf > results.txt
# Then type 'book'
# The results are displayed in a list with one line highlighted.
# Press <ENTER> to save that line into 'results.txt' and end the fzf
↪  session
```

3.2 Select Command History

This example will display your command history and allow you to search through it using fzf. You can then select a command and press enter to execute it.

```
history | fzf
```

3.3 Edit the Result in One Go

This example will open the file you select in `vim`. You can use `fzf` to quickly search and select the file you want to open.

```
# This will search all files below the current directory
# The selected file will be opened in vim
fzf --print0 | xargs -0 -o vim
# The same, shorter, but less robust would be
vim $(fzf)
```

3.4 Chaining patterns

You can use the interactive technique above to search for complex things.

1. Find a file that contains exactly the word 'book'
2. And is a Markdown file
3. And the path starts with 'scratch'

You would enter interactively

```
'book .md$ ^scratch
```

This could for example give file `scratch/todo/books-to-write.md`.

fzf	Kind of matching	Notes
smthng	fuzzy-match	Items that match smthng
'calm	exact-match (quoted)	Items that include calm
^movie	prefix-exact-match	Items that start with movie
.pdf$	suffix-exact-match	Items that end with .pdf
!water	inverse-exact-match	Items that do not include water
!^movie	inverse-prefix-exact-match	Items that do not start with movie
!.pdf$	inverse-suffix-exact-match	Items that do not end with .pdf

3.5 Change Directories

This example should not be taken too serious. This command will allow you to quickly navigate your file system using `fzf`. It will list all directories under the current directory and allow you to select one to change into. However it also will display dot-files. So that is basically useless in `~/` or a git repository.

```
cd $(find . -type d -print 2>/dev/null | fzf)
```

This improved example is similar to the previous one, but it uses the -not -path `*/\.*` option to exclude directories that begin with a dot (i.e., dot-files). This ensures that only visible directories are displayed in the `fzf` interface, making it easier to navigate the file system.

```
cd $(find . -type d -not -path '*/\.*' -print 2>/dev/null | fzf)
```

3.6 Select Git Commit and Show the Commit

```
git show "$(git log --oneline | fzf | awk '{print $1}')"
```

This command shows the list of Git commits in the current repository using `git log --oneline`, filters them using `fzf`, extracts the commit hash using `awk`, and then displays the selected commit using `git show`.

4 History

Version	Date	Notes
0.1.2	2023-04-12	Add git example
0.1.1	2023-03-30	Improve writing, add 2 examples
0.1.0	2023-02-16	Initial release

5 Disclaimer of Warranty

6 Limitation of Liability