Your Laptop is not a Bop-It!

•••

Leveraging the Power of Your Keyboard to Improve the Developer Experience

Geoff Cureton

@gpcureton@mstdn.social

https://bopit.fandom.com/wiki/Bop_It_Extreme

Overview

- My Origin Story
- Minimising Cognitive Load
  - Economy of Movement
  - Economy of Search
- Questions

Part I: My Origin Story

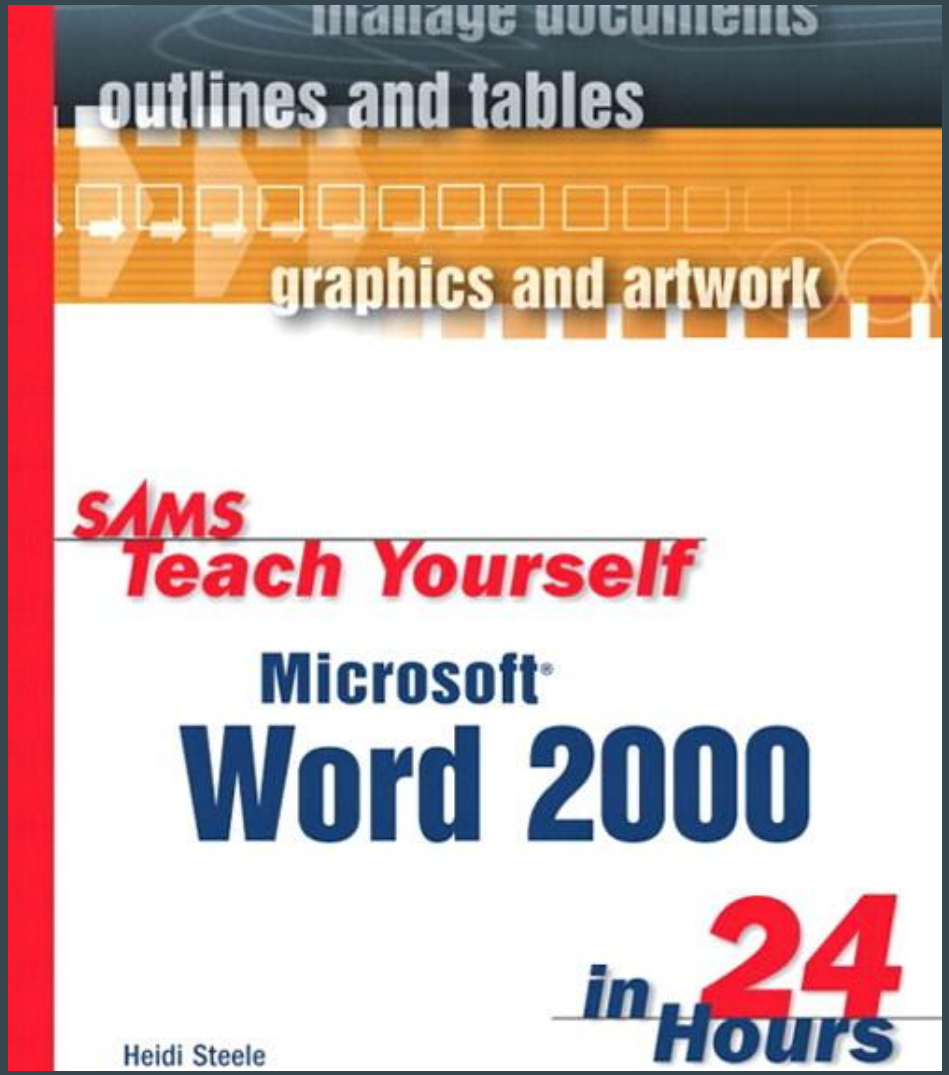
My Origin Story

- Undergrad in Physics (‘94 - ‘96)
  - Windows 3.1
  - MS Word / Excel
  - Mouse Centered Workflow
  - Floppy Disks

West Nevada Community College, 199?

My Origin Story

- Undergrad in Physics (‘94 - ‘96)
  - Windows 3.1
  - MS Word / Excel
  - Mouse Centered Workflow
  - Floppy Disks
- Honours Year (1997): When I Try To Get Serious...


manage documents
outlines and tables
graphics and artwork
SAMS
Teach Yourself
Microsoft®
Word 2000
in 24 Hours
Heidi Steele

… but then I met this guy…

BeeWare About Documentation Community Contributing News Donate Edit on GitHub

Home > About BeeWare

The Bee Team

The core team behind BeeWare development

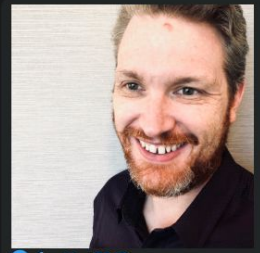
So who are the people behind BeeWare? Well, there's a huge group of contributors, but the project is managed by the Bee Team.

Current team

Russell Keith-Magee

Dr Russell Keith-Magee is the founder of the BeeWare project. He joined the Django core team in 2006, and for 5 years, was President of the Django Software Foundation. He has been a member of the CPython core team since May 2024.

Russell is a Principal Software Engineer in the Open Source group at Anaconda (a BeeWare Gold sponsor).



freakboy3742
@freakboy3742@cloudisland.nz
freakboy3742
russell@beeware.org
Superpower: Bee-nevolent Dictator, For Now

... who “ruined” my life by introducing me to LaTeX 😵

```latex
\section{Simulation of Linear Surfaces\label{sec:Linear_sim}}

Since we desire to simulate ocean surfaces in the spectral domain, our simulation
method can be based around one of two representations of the linear, Gaussian
process \cite{Rice54}:
\begin{equation}
\label{eqn:linear_one}
\eta (x)=\sum ^{N}_{n=1}a_{n}\cos (k_{n}x)+b_{n}\sin (k_{n}x)\, ,
\end{equation}


or
\begin{equation}
\label{eqn:linear_two}
\eta (x)=\sum ^{N}_{n=1}c_{n}\cos (k_{n}x+\phi _{n})\, .
\end{equation}


In (\ref{eqn:linear_one}), \( a_{n} \) and \( b_{n} \) are independent random,
Gaussian variables, with zero mean and variance
\[
\sigma ^{2}_{n}=\Psi (k_{n})\Delta k\, ,\]


while in (\ref{eqn:linear_two}) the amplitudes \( c_{n} \) are given by
\[
c_{n}=\sqrt{2\Psi (k_{n})\Delta k}\, ,\]
```

2 Simulation of Linear Surfaces

Since we desire to simulate ocean surfaces in the spectral domain, our simulation method can be based around one of two representations of the linear, Gaussian process (Rice 1954):

$$\eta(x) = \sum_{n=1}^{N} a_n \cos(k_n x) + b_n \sin(k_n x)\,, \tag{1}$$

or

$$\eta(x) = \sum_{n=1}^{N} c_n \cos(k_n x + \phi_n)\,. \tag{2}$$

In (1), a_n and b_n are independent random, Gaussian variables, with zero mean and variance

$$\sigma_n^2 = \Psi(k_n)\Delta k\,,$$

while in (2) the amplitudes c_n are given by

$$c_n = \sqrt{2\Psi(k_n)\Delta k}\,,$$

But seriously... why was exposure to LaTeX significant?

- It introduced me to the idea that a visual element and its representation/implementation can be decoupled (like HTML, Markdown...)

But seriously... why was exposure to LaTeX significant?

- It introduced me to the idea that a visual element and its representation/implementation can be decoupled (like HTML, Markdown...)

- Also...

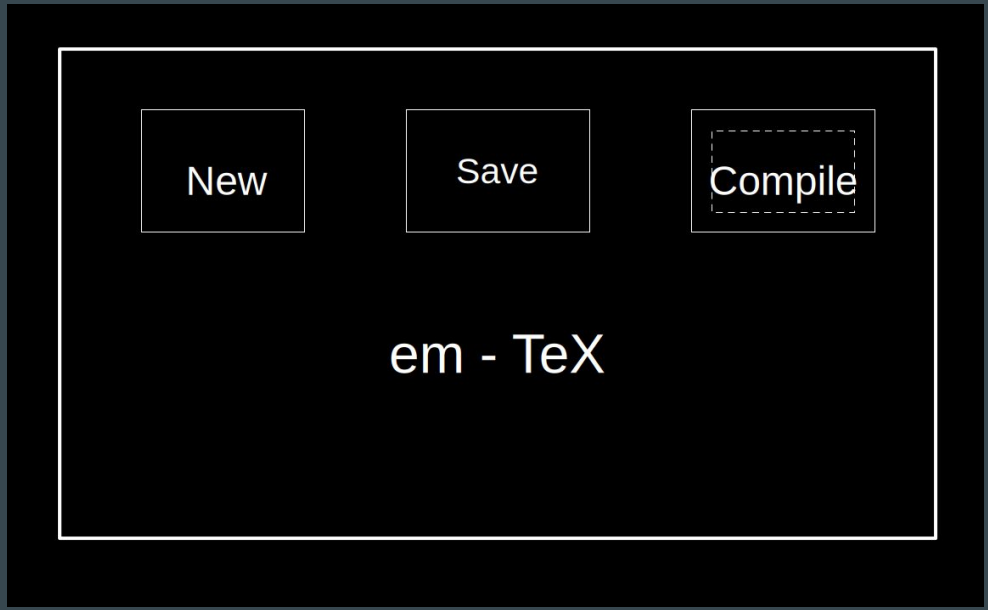
But seriously... why was exposure to LaTeX significant?

- It introduced me to the idea that a visual element and its representation/implementation can be decoupled (like HTML, Markdown...)

- Also... LaTeX is cool.

But Also...

- I ran my LaTeX “distribution” em-TeX using a little GUI called “emTeXGI”.
- Sadly, emTeXGI has been memory–holed; no images of it remain, but it looked something like this...



But Also...

- I ran my LaTeX “distribution” em-TeX using a little GUI called “emTeXGI”.
- Sadly, emTeXGI has been memory–holed; no images of it remain, but it looked something like this...

- Alt-Tab between GUI and source
- Tab between “Save”, “Compile”

- I became irate when I couldn’t do this.

The Interregnum

- Eventually I went to grad school, and intermittently used Vim (oooh, foreshadowing!), but not in a serious way.
- Various jobs required me to use Windows, Visual Studio.NET, and other mousey things.
- This situation persisted for good while, punctuated by my occasional insistence on using LaTeX to make presentations (I'm cured of that), and conference posters (of which I'm still afflicted).
- Then came...

The Pandemic!

The Pandemic

- While I had tinkered around with keyboard-based workflows previously, I hadn't seriously put any time into it.
- By the middle of 2020, I found that I was spending a lot more time on my computer (for good or ill), and was itching to do something different.
- A few diversions started to take a more important role:
  - Coding YouTubers
  - Coding YouTubers who use Vim
  - Mechanical Keyboards
- Combining these interests led me to...

Part II: Minimizing Cognitive Load

Minimizing Cognitive Load

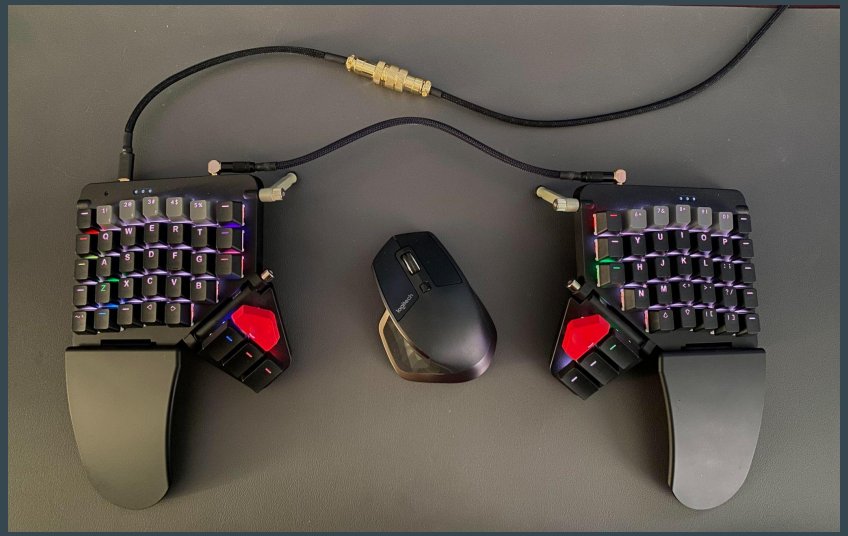
- Reducing behaviours and actions which interrupt your current task. People are bad multitaskers. They just are.
- The things I do to streamline my workflow fall into two broad categories:

    - Economy of Movement
    - Economy of Search

Economy of Movement

- This is where the concept of the Bop-It comes in: you will be more effective if you are not flailing away at your keyboard. Or between the keyboard and mouse.
- Use your mouse when you need to, and your keyboard when you don't
- For just the keyboard, economy of movement can be achieved either...

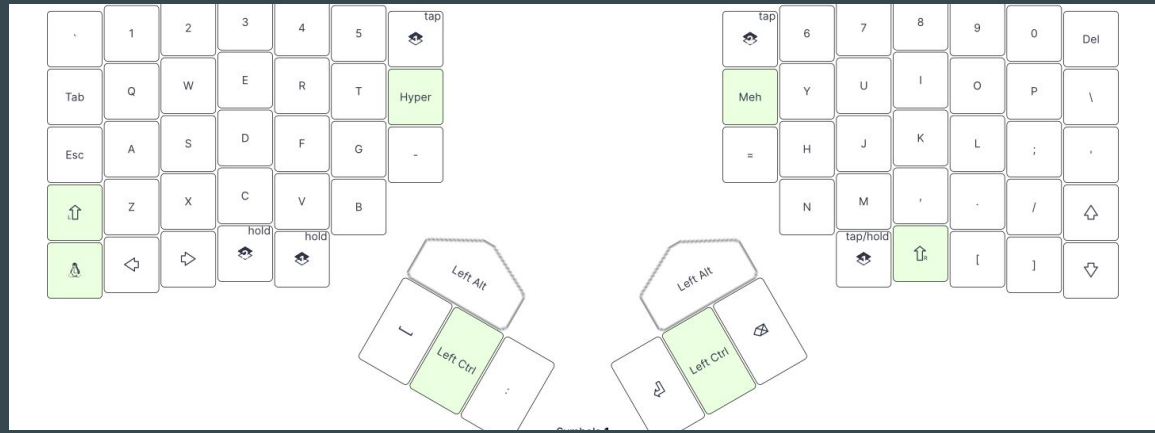
- Hardware (keyboards)



Economy of Movement

- This is where the concept of the Bop-It comes in: you will be more effective if you are not flailing away at your keyboard. Or between the keyboard and mouse.
- Use your mouse when you need to, and your keyboard when you don't
- For just the keyboard, economy of movement can be achieved either...

- Or Software (key-mappings)



Keyboards

- I don’t get too invested in the notion of and “ergonomic” keyboard. I’m sure it’s a thing, but 🤷
- Split keyboards (as on the previous slides) are fine, but a good layout can be just as good.
- My most used keyboard...

Keyboards

- I don't get too invested in the notion of and "ergonomic" keyboard. I'm sure it's a thing, but 🤷
- Split keyboards (as on the previous slides) are fine, but a good layout can be just as good.
- My most used keyboard...

Keyboards

- Orthocolumnar! (keys are in vertical rows, not staggered)
- This is a 40% keyboard (w.r.t. a full-size keyboard's number of keys)
- How do we cram all the functionality we want into such a small size...

Keyboards

- Orthocolumnar! (keys are in vertical rows, not staggered)
- This is a 40% keyboard (w.r.t. a full-size keyboard's number of keys)
- How do we cram all the functionality we want into such a small size...
- LAYERS!

Keyboards

- You already use layers: the SHIFT key takes you to the “CAPS and SYMBOLS” layer.
- This keyboard has layer “raise” and “lower” keys either side of the space bar.
- LED lighting of the keys are used as a visual cue

Keyboards

- The “lower” layer converts the top row to the usual symbols you’d find along the top of a keyboard
- The “< > ⬇️⬆️” keys are Line start/end, Page down/up respectively.
- “hjkl” are left/down/up/right respectively (more foreshadowing!)

Keyboards

- The “raise” layer converts the top row to the usual numbers.
- Makes an upside-down keypad: QWEASDZXCV➡ 1-9,0
- “Keypad” is purple, as a visual cue

Keyboards

- Both layer keys simultaneously activates the “parenthesis” layer
- “RU”, “FJ” and “VM” give the “()”, “[]” and “{}” pairs respectively. Note the symmetry...
- “D” and “K” give “^” and “$” (IYKYK)

Hot Keys

- Sometimes you don't have your fancy programmable keyboard with you, what do you do then? Hotkey daemons!
- Sxhkd (https://github.com/baskerville/sxhkd), Linux
- Kanata (https://github.com/jtroo/kanata), Linux
- Skhd (https://github.com/asmvik/skhd), MacOS

```
#     MULTIMEDIA KEYS
##########################################################################

#Raises volume
XF86AudioRaiseVolume
    amixer set Master 10%+

#Lowers volume
XF86AudioLowerVolume
    amixer set Master 10%-

#Mute
XF86AudioMute
```

Hot Keys

- Key mapping daemons are generally used to create mappings which are independent of the OS or desktop environment being used
- In the example below, the laptop media keys are being mapped to a particular audio mixer app.

```
#     MULTIMEDIA KEYS
#############################################################################

#Raises volume
XF86AudioRaiseVolume
    amixer set Master 10%+

#Lowers volume
XF86AudioLowerVolume
    amixer set Master 10%-

#Mute
XF86AudioMute
```

Web Browsing

- Navigating to links on a webpage might seem like “search”, but you already have to find the link with your eyes, so it’s just movement.

Web Browsing

- Navigating to links on a webpage might seem like “search”, but you already have to find the link with your eyes, so it’s just movement.
- Shown below is “Qutebrower”, “f” for “find” brings up key combinations to select the various links. Note that each of the character pairs have either characters accessible from either hand, or by a comfortable rolling motion of the same hand. No awkward stretches.

Economy of Search

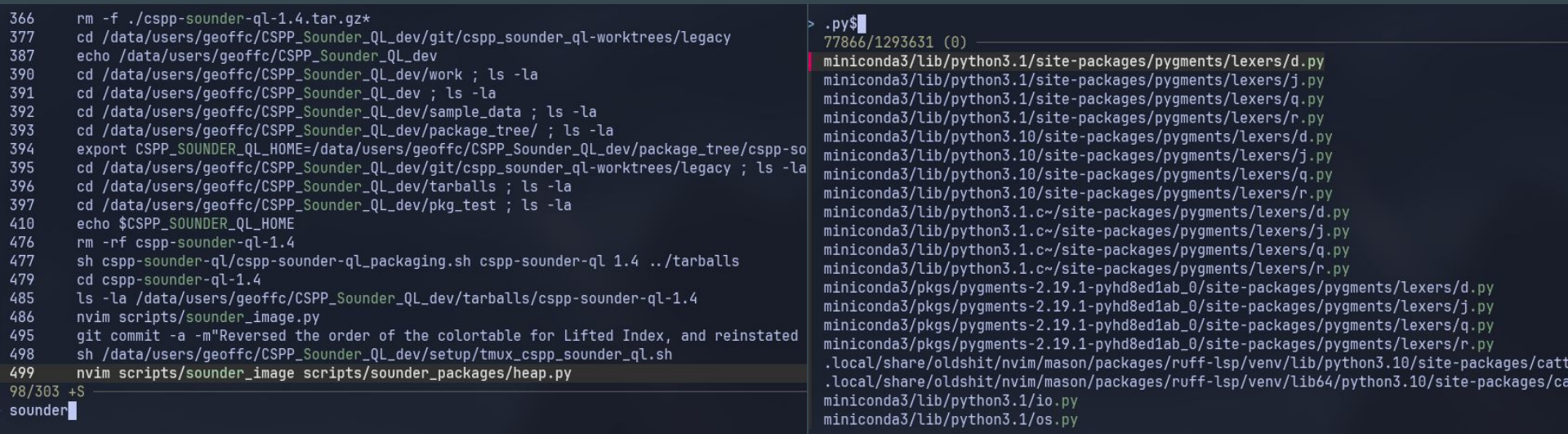
- Optimizing the process of searching can also be thought of as optimizing navigation: if we're looking for something, we'd rather be able to teleport there rather than walk.
- We can be searching for a particular string in a text file, or a workspace on our desktop, with the aim of going there
- Rather than inspecting a large list/table to find the thing we want (and clicking on it), it is preferable to type a few characters to zero in on what we want.

Fuzzy Finders

- The notion of “fuzzy finding” can be applied to a range of different contexts:
  - finding the app you want to load
  - finding a file on your file system
  - finding a string in a text file
- The utility of fuzzy finding depends on at least having some idea of what you are looking for, i.e.: part of the name
- There are some instances where a text-based search my be slower that just using your Mark-1 Eyeball to directly find the thing: they will be discussed as we get to them

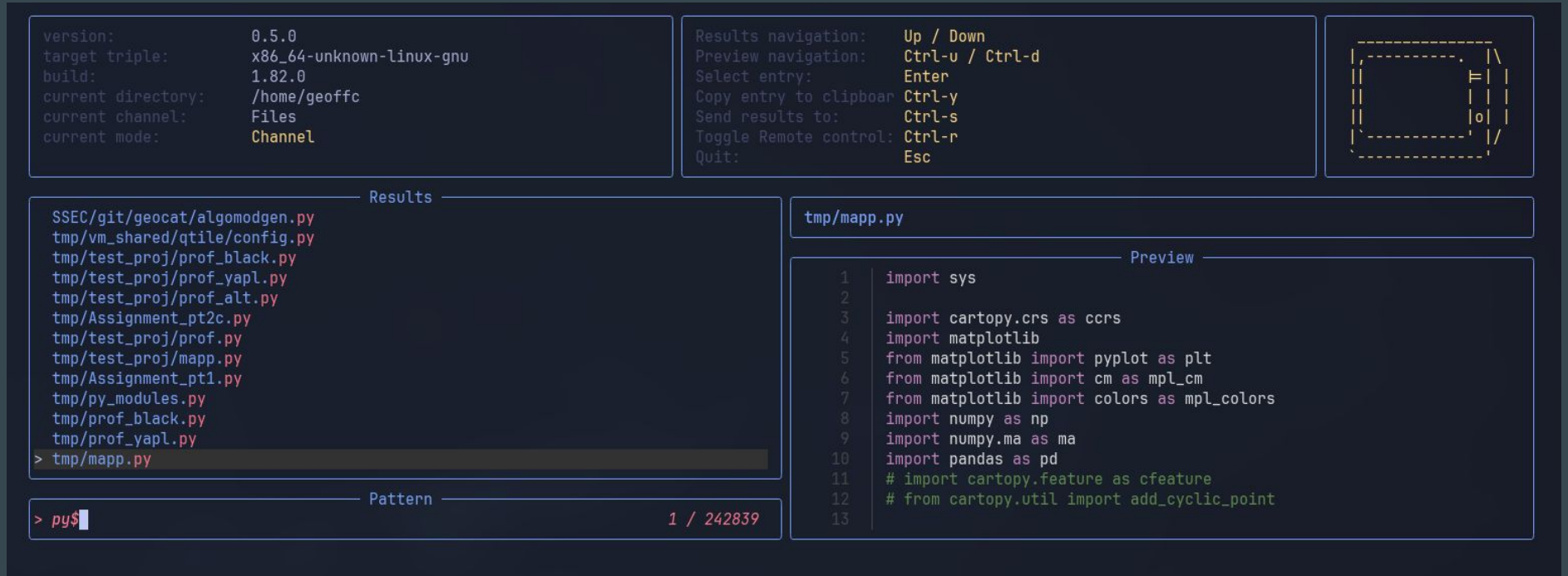
FZF (https://github.com/junegunn/fzf)

- A fuzzy finder for your command line
- Can search your command history, or for a file on your filesystem (and many things besides).
- CTRL-R and CTRL-T give history and filesystem fuzzy finding respectively



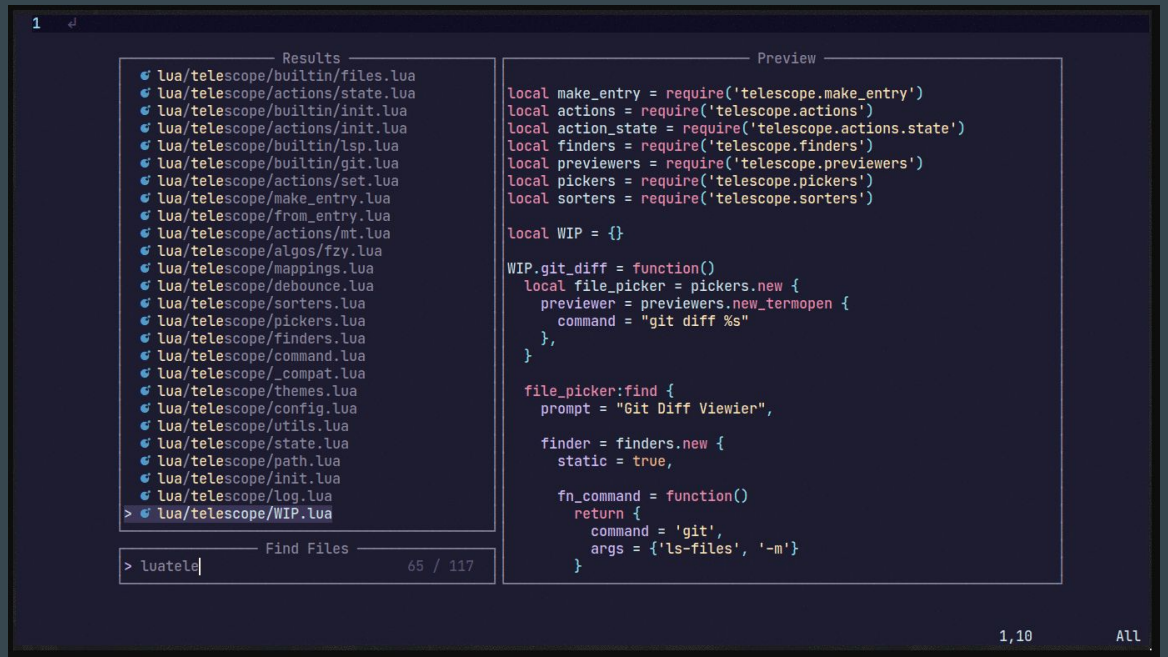
television (https://github.com/alexpasmantier/television)

- “A very fast, portable and hackable fuzzy finder for the terminal.”
- Can search filesystem, code, processes, git repos, env vars, docker containers...



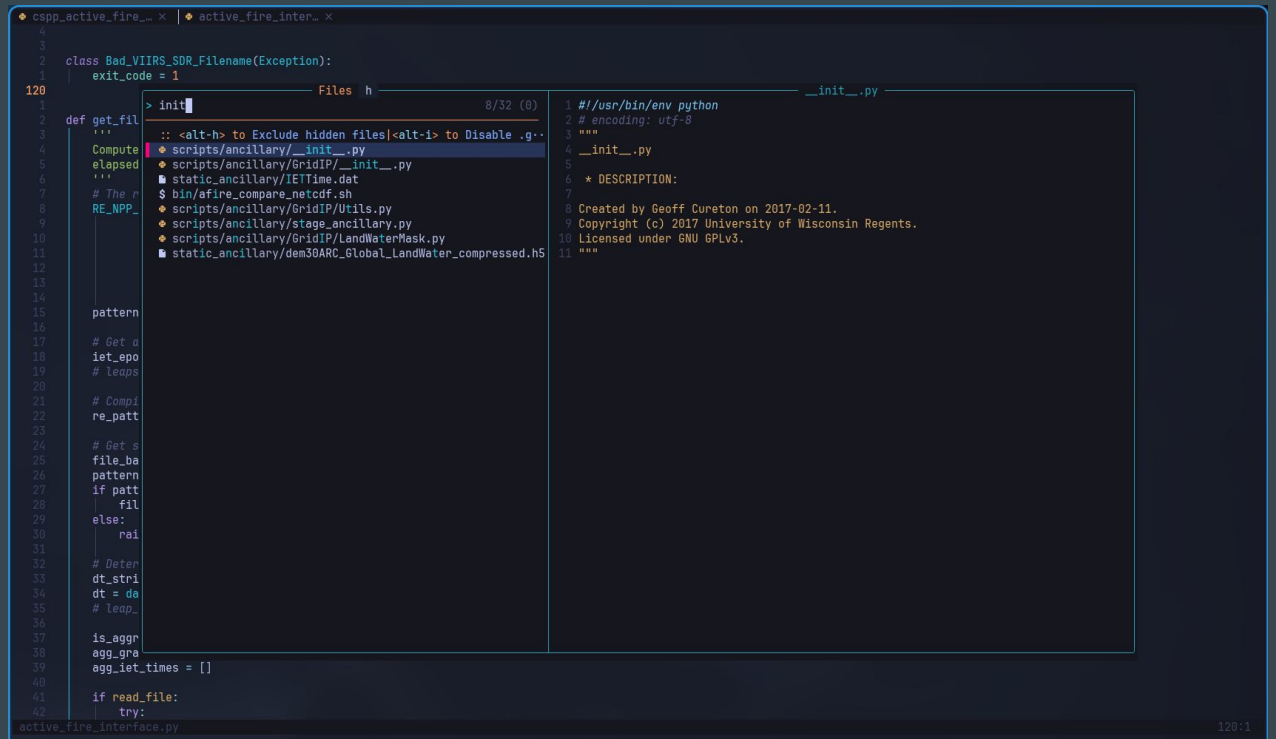
Telescope (https://github.com/nvim-telescope/telescope.nvim)

- The first general-purpose fuzzy finder plugin for NeoVim.
- You could add your own finders (say, for your Markdown notes)



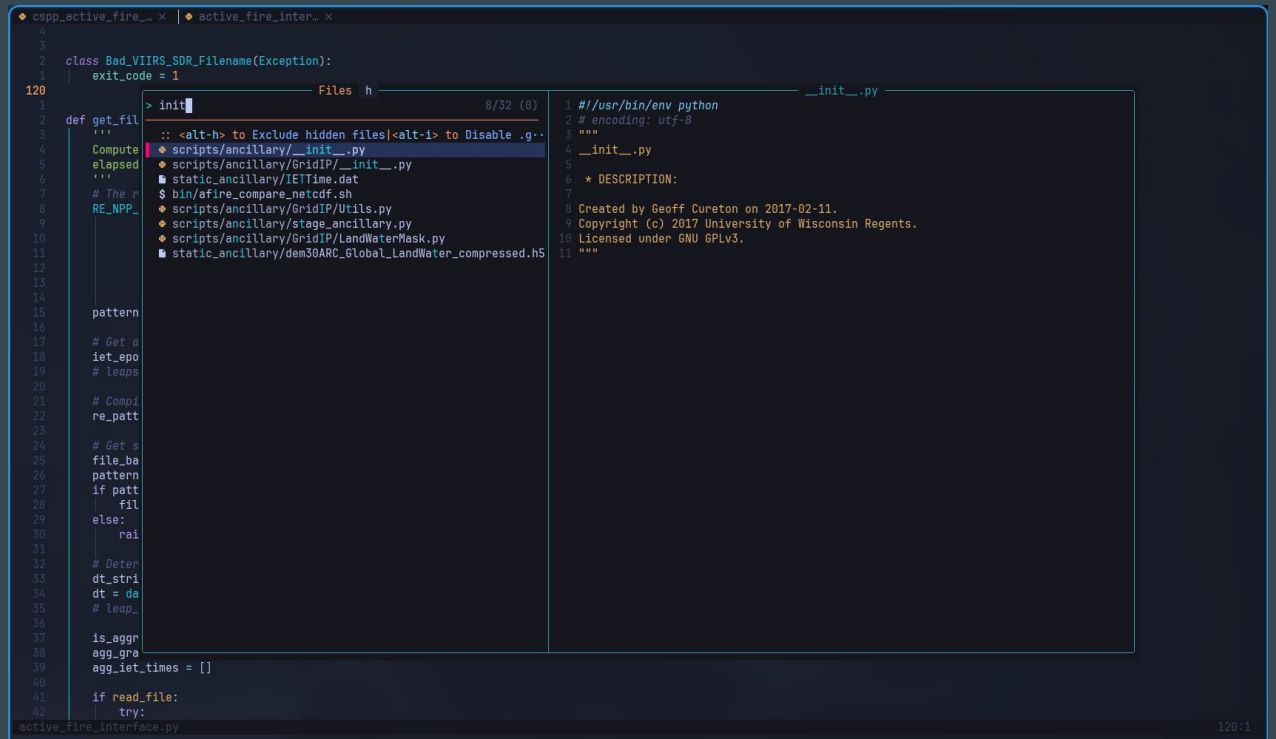
LazyVim (https://www.lazyvim.org/)

- A NeoVim “distribution” which provides a range of different search modes....



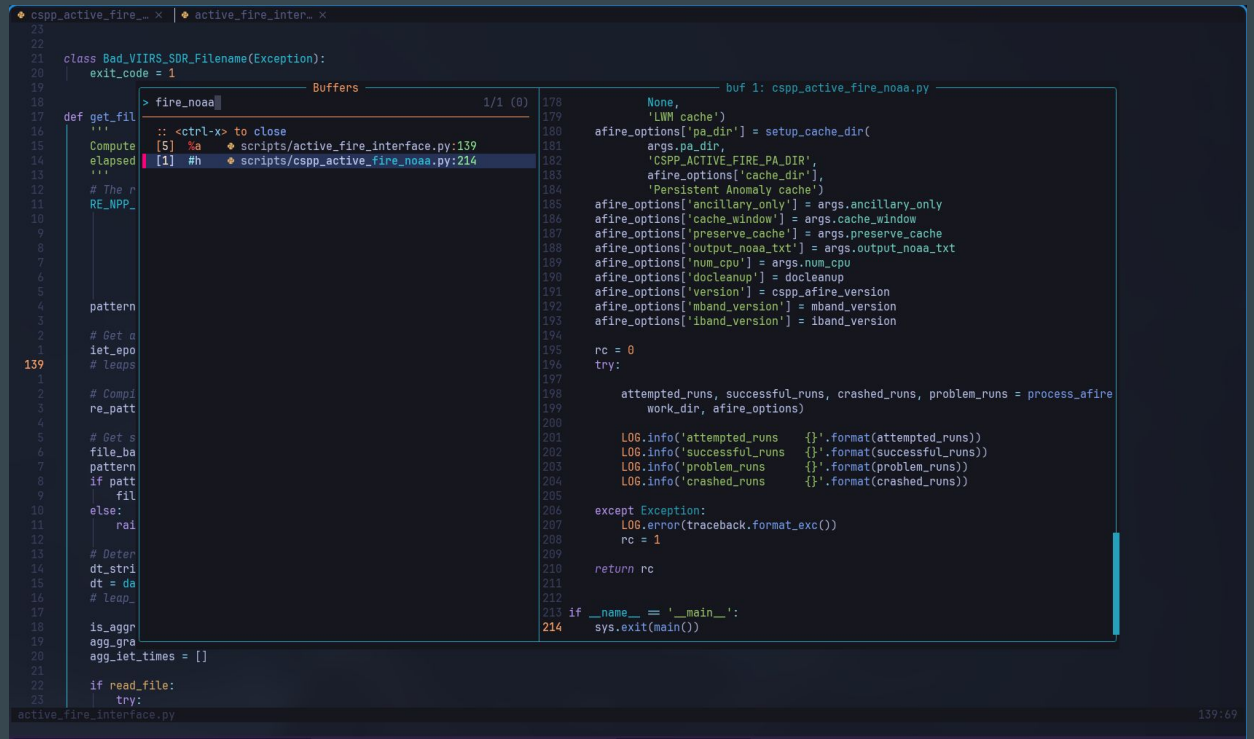
LazyVim (https://www.lazyvim.org/)

- Files



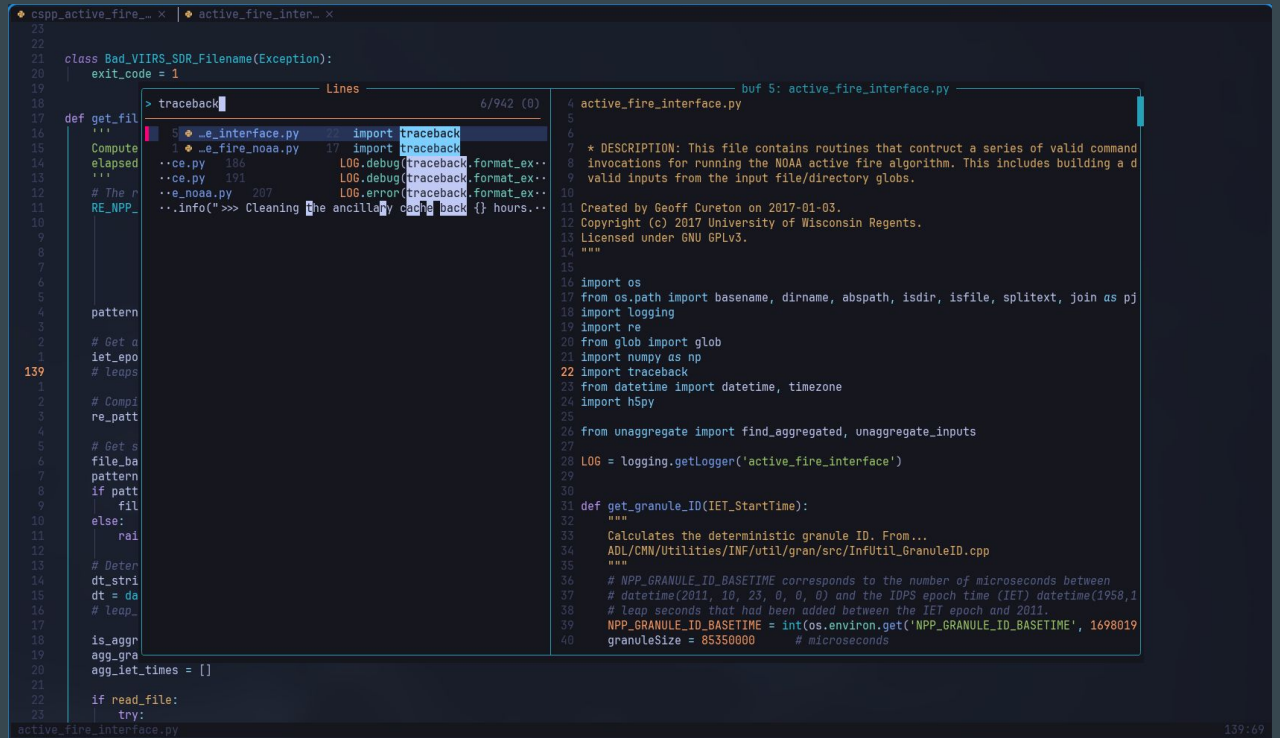
LazyVim (https://www.lazyvim.org/)

- Buffers



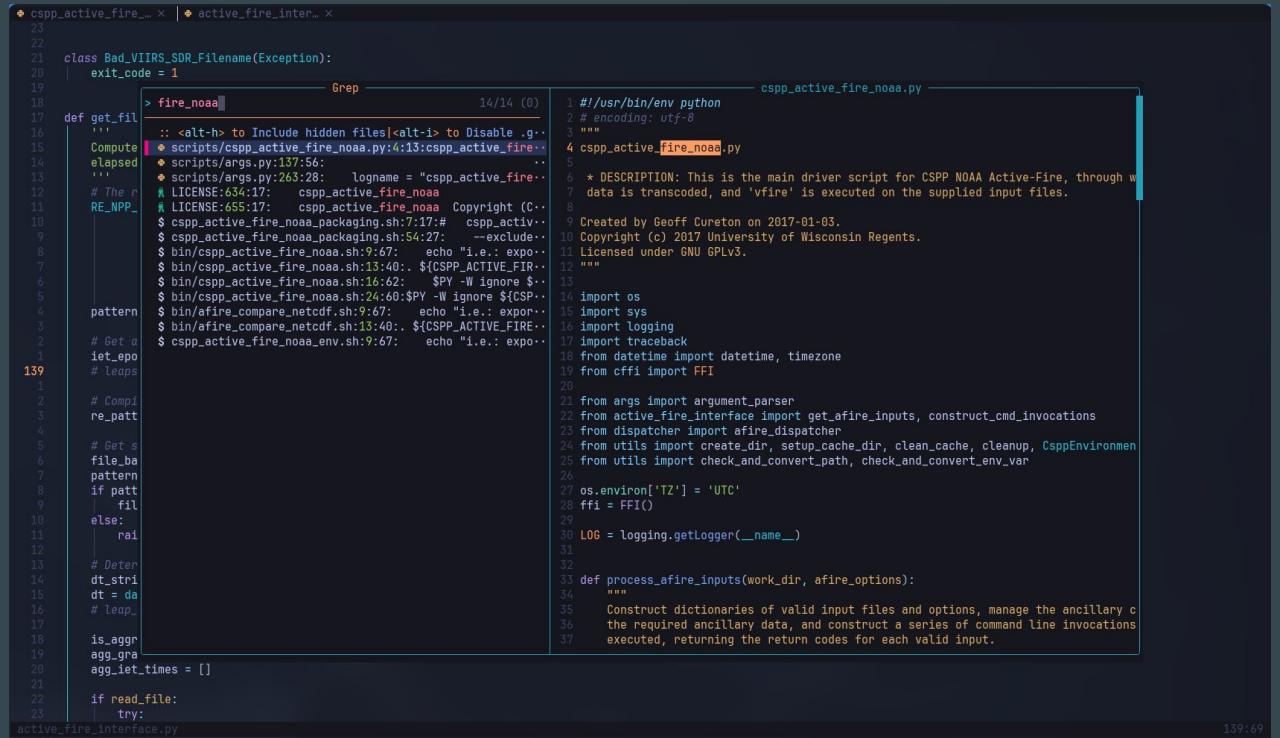
LazyVim (https://www.lazyvim.org/)

- Search within current buffer



LazyVim (https://www.lazyvim.org/)

- Grep files in CWD (with ripgrep!)

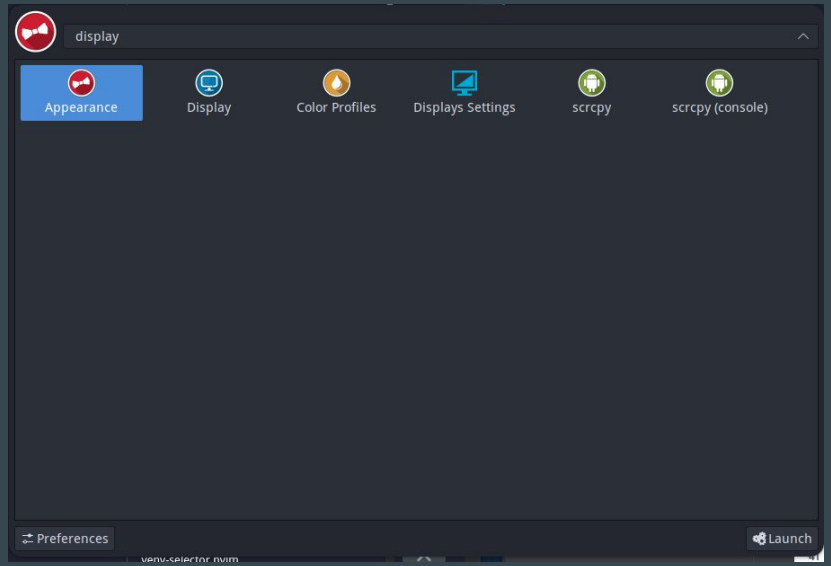


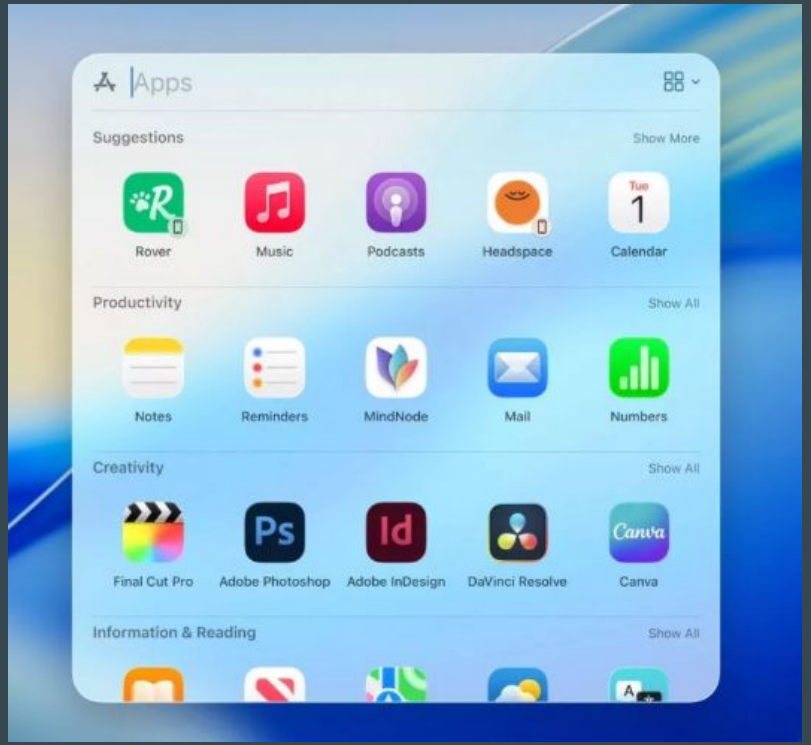
Harpoon (https://github.com/ThePrimeagen/harpoon)

- When working in a large codebase, it's common to need to jump back and forth between one of a few points (say, a function usage and its definition)
- Harpoon is predicated on the idea that people cannot meaningfully use more than several points of context at a time.
- Each of those points can be made easily accessible using a hotkey: <something>-1, <something>-2 etc...
- Using these limited number of mappings, one can rapidly navigate code relevant to a task. As the task changes, so do the mappings.
- This workflow is an alternative to trying to have all of your context on screen at once... bigger monitors!, more monitors!

Application Pickers

- Beats navigating down a menu tree!



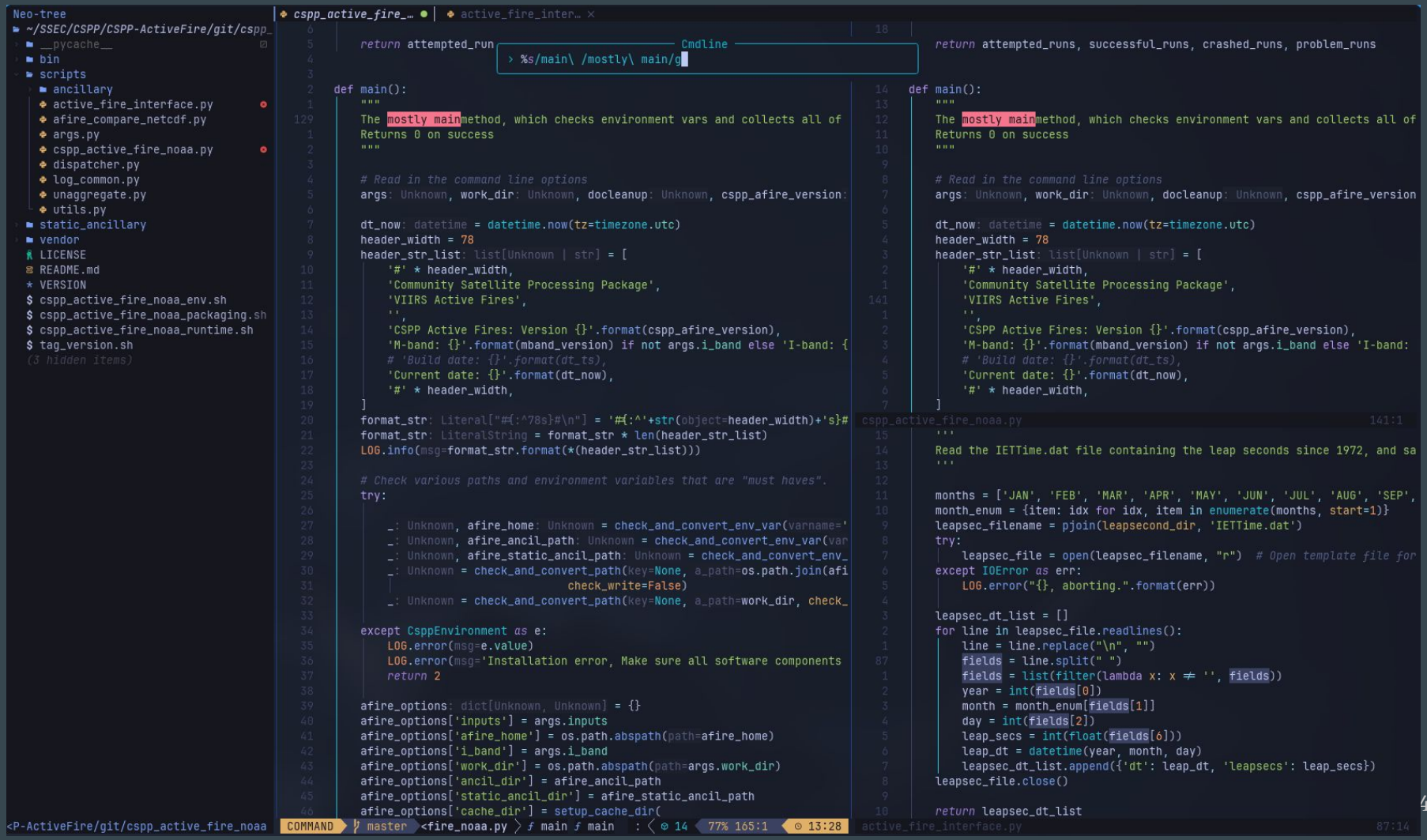


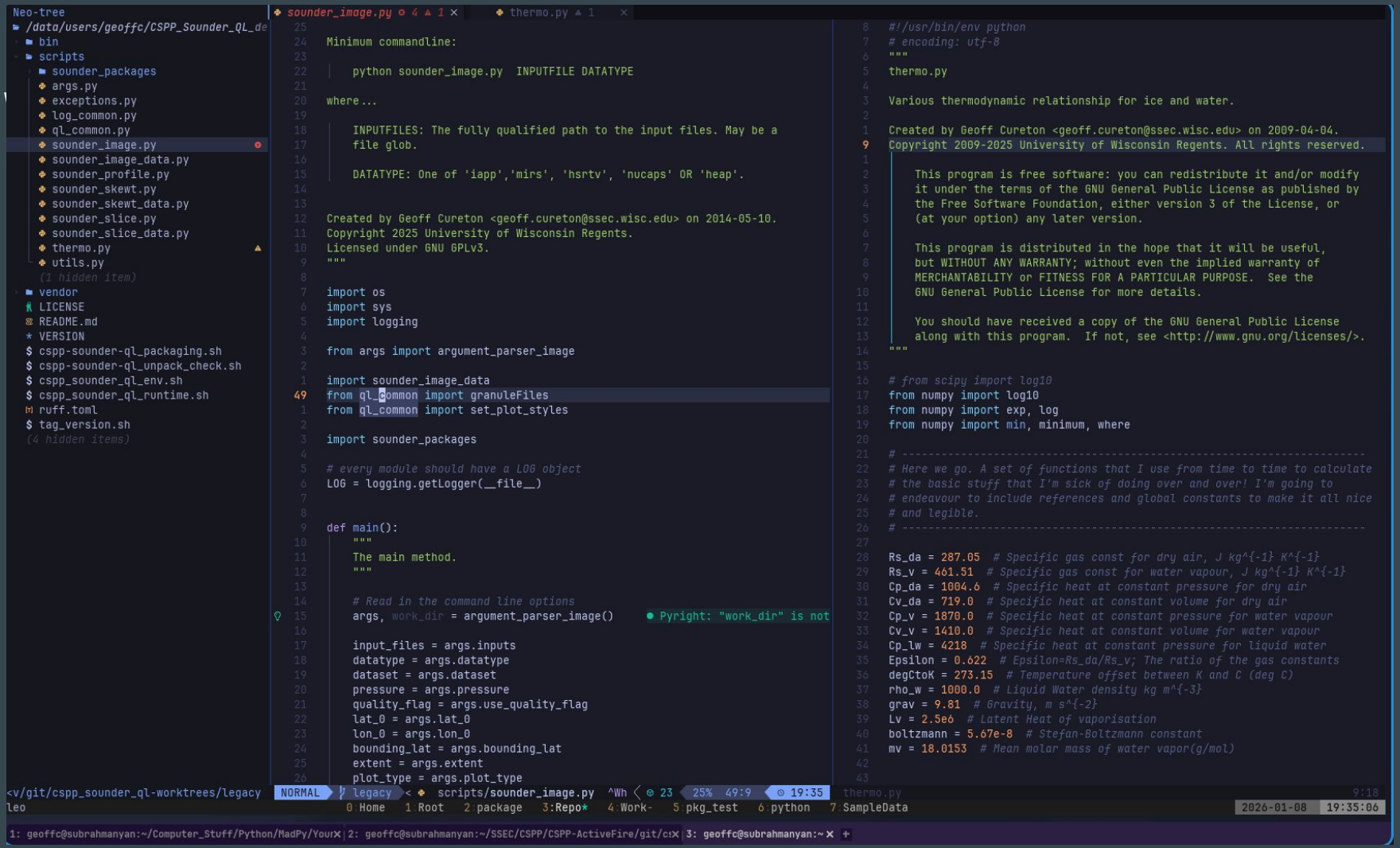
Miscellaneous

- There are some parts of my workflow that don't fit neatly a particular category.
- I'll just go through them and discuss why I find them useful

Vim / NeoVim / Emacs + TMUX

- A text editor with useful keybinds to navigate between the various elements (buffers, windows, dialogs, ...)
- Many editors outside of the main TUI based ones have plugins to replicate their behavior (VS Code - VIM keybinds)
- A terminal-based editor can be run “locally” on a remote server (inside a Screen/TMUX/Zellij session), you can log out, go home, and pick up where you left off.

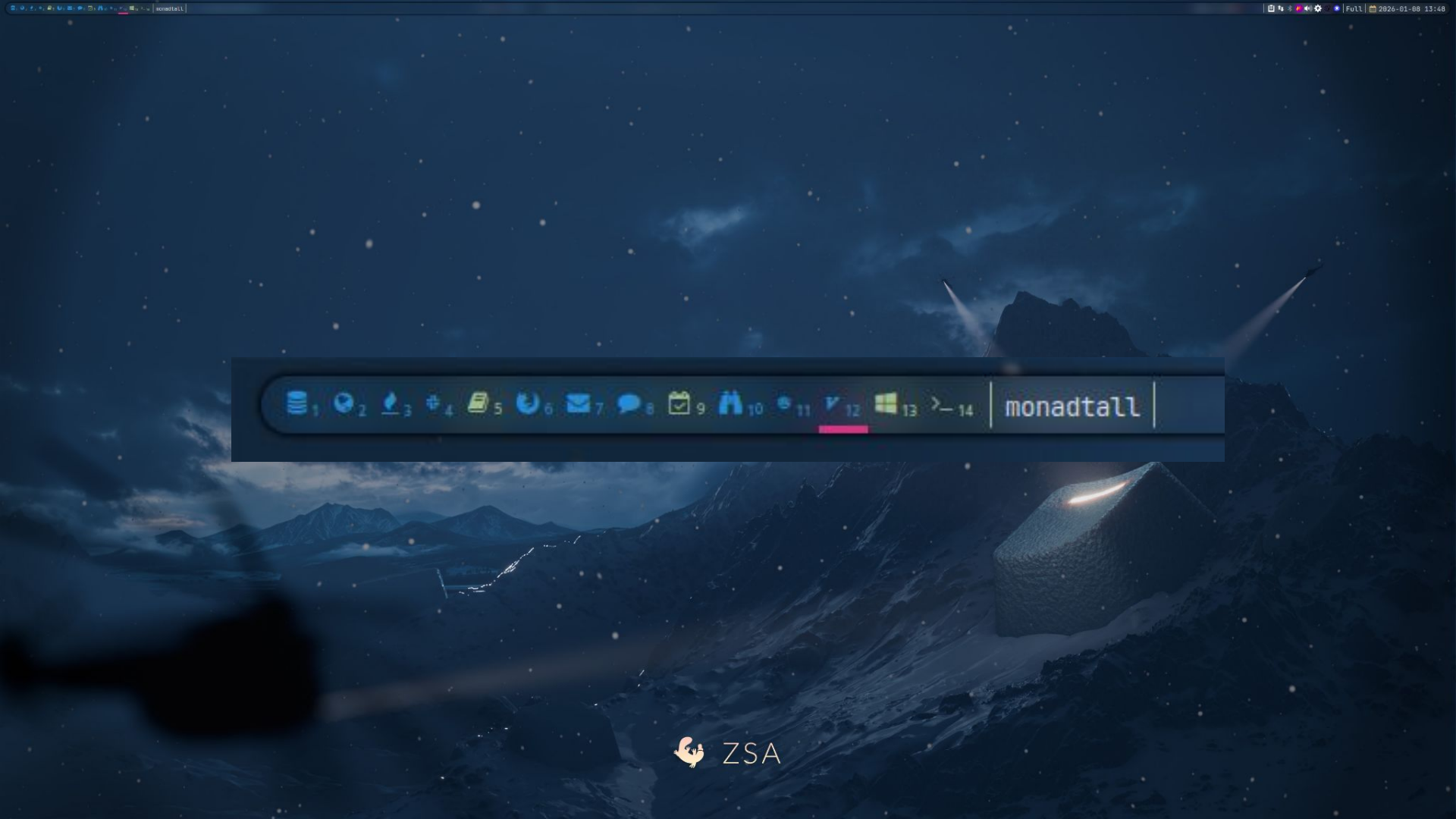

Neo-tree
~/SSEC/CSPP/CSPP-ActiveFire/git/cspp_
__pycache__
bin
scripts
ancillary
active_fire_interface.py
afire_compare_netcdf.py
args.py
cspp_active_fire_noaa.py
dispatcher.py
log_common.py
unaggregate.py
utils.py
static_ancillary
vendor
LICENSE
README.md
VERSION
cspp_active_fire_noaa_env.sh
cspp_active_fire_noaa_packaging.sh
cspp_active_fire_noaa_runtime.sh
tag_version.sh
(3 hidden items)
cspp_active_fire_…
active_fire_inter…
Cmdline
%s/main\ /mostly\ main/g
return attempted_run
def main():
"""
The mostly mainmethod, which checks environment vars and collects all of
Returns 0 on success
"""
Read in the command line options
args: Unknown, work_dir: Unknown, docleanup: Unknown, cspp_afire_version:
dt_now: datetime = datetime.now(tz=timezone.utc)
header_width = 78
header_str_list: list[Unknown | str] = [
'#' * header_width,
'Community Satellite Processing Package',
'VIIRS Active Fires',
'',
'CSPP Active Fires: Version {}'.format(cspp_afire_version),
'M-band: {}'.format(mband_version) if not args.i_band else 'I-band: {
# 'Build date: {}'.format(dt_ts),
'Current date: {}'.format(dt_now),
'#' * header_width,
]
format_str: Literal["#{:^78s}#\n"] = '#{:^'+str(object=header_width)+'s}#
format_str: LiteralString = format_str * len(header_str_list)
LOG.info(msg=format_str.format(*(header_str_list)))
# Check various paths and environment variables that are "must haves".
try:
_: Unknown, afire_home: Unknown = check_and_convert_env_var(varname='
_: Unknown, afire_ancil_path: Unknown = check_and_convert_env_var(var
_: Unknown, afire_static_ancil_path: Unknown = check_and_convert_env_
_: Unknown = check_and_convert_path(key=None, a_path=os.path.join(afi
check_write=False)
_: Unknown = check_and_convert_path(key=None, a_path=work_dir, check_
except CsppEnvironment as e:
LOG.error(msg=e.value)
LOG.error(msg='Installation error, Make sure all software components
return 2
afire_options: dict[Unknown, Unknown] = {}
afire_options['inputs'] = args.inputs
afire_options['afire_home'] = os.path.abspath(path=afire_home)
afire_options['i_band'] = args.i_band
afire_options['work_dir'] = os.path.abspath(path=args.work_dir)
afire_options['ancil_dir'] = afire_ancil_path
afire_options['static_ancil_dir'] = afire_static_ancil_path
afire_options['cache_dir'] = setup_cache_dir(
<P-ActiveFire/git/cspp_active_fire_noaa
COMMAND master <fire_noaa.py f main f main : 14 77% 165:1 13:28
return attempted_runs, successful_runs, crashed_runs, problem_runs
def main():
"""
The mostly mainmethod, which checks environment vars and collects all of
Returns 0 on success
"""
Read in the command line options
args: Unknown, work_dir: Unknown, docleanup: Unknown, cspp_afire_version
dt_now: datetime = datetime.now(tz=timezone.utc)
header_width = 78
header_str_list: list[Unknown | str] = [
'#' * header_width,
'Community Satellite Processing Package',
'VIIRS Active Fires',
'',
'CSPP Active Fires: Version {}'.format(cspp_afire_version),
'M-band: {}'.format(mband_version) if not args.i_band else 'I-band:
'Build date: {}'.format(dt_ts),
'Current date: {}'.format(dt_now),
'#' * header_width,
]
cspp_active_fire_noaa.py 141:1
'''
Read the IETTime.dat file containing the leap seconds since 1972, and sa
'''
months = ['JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP',
month_enum = {item: idx for idx, item in enumerate(months, start=1)}
leapsec_filename = pjoin(leapsecond_dir, 'IETTime.dat')
try:
leapsec_file = open(leapsec_filename, "r") # Open template file for
except IOError as err:
LOG.error("{}, aborting.".format(err))
leapsec_dt_list = []
for line in leapsec_file.readlines():
line = line.replace("\n", "")
fields = line.split(" ")
fields = list(filter(lambda x: x != '', fields))
year = int(fields[0])
month = month_enum[fields[1]]
day = int(fields[2])
leap_secs = int(float(fields[6]))
leap_dt = datetime(year, month, day)
leapsec_dt_list.append({'dt': leap_dt, 'leapsecs': leap_secs})
leapsec_file.close()
return leapsec_dt_list
active_fire_interface.py 87:14

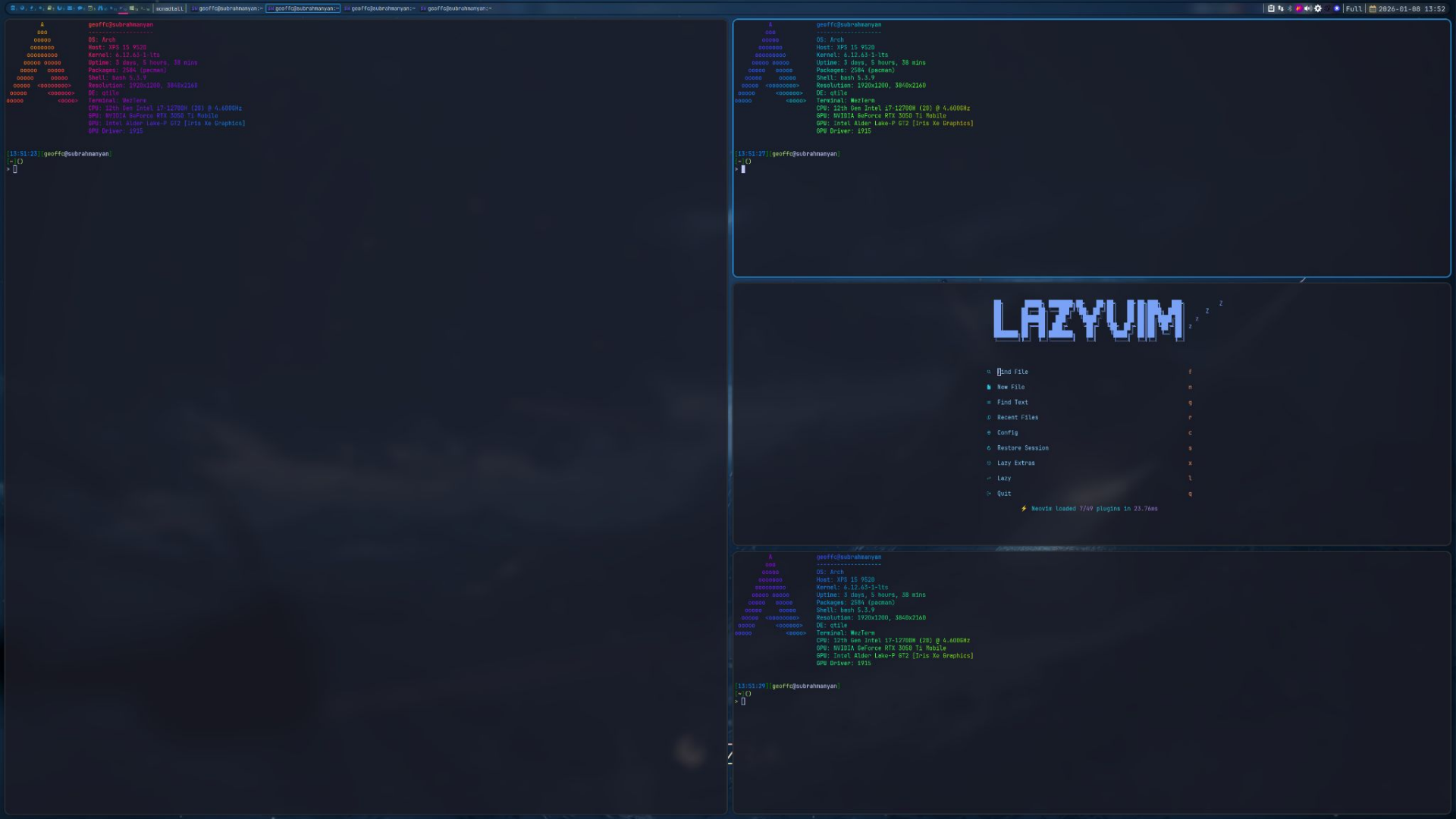

Neo-tree
/data/users/geoffc/CSPP_Sounder_QL_de
bin
scripts
sounder_packages
args.py
exceptions.py
log_common.py
ql_common.py
sounder_image.py
sounder_image_data.py
sounder_profile.py
sounder_skewt.py
sounder_skewt_data.py
sounder_slice.py
sounder_slice_data.py
thermo.py
utils.py
(1 hidden item)
vendor
LICENSE
README.md
VERSION
$ cspp-sounder-ql_packaging.sh
$ cspp-sounder-ql_unpack_check.sh
$ cspp_sounder_ql_env.sh
$ cspp_sounder_ql_runtime.sh
ruff.toml
$ tag_version.sh
(4 hidden items)
sounder_image.py 4 1
thermo.py 1
Minimum commandline:
python sounder_image.py INPUTFILE DATATYPE
where...
INPUTFILES: The fully qualified path to the input files. May be a file glob.
DATATYPE: One of 'iapp','mirs', 'hsrtv', 'nucaps' OR 'heap'.
Created by Geoff Cureton <geoff.cureton@ssec.wisc.edu> on 2014-05-10.
Copyright 2025 University of Wisconsin Regents.
Licensed under GNU GPLv3.
"""
import os
import sys
import logging
from args import argument_parser_image
import sounder_image_data
from ql_common import granuleFiles
from ql_common import set_plot_styles
import sounder_packages
# every module should have a LOG object
LOG = logging.getLogger(__file__)
def main():
"""
The main method.
"""
Read in the command line options
args, work_dir = argument_parser_image()
Pyright: "work_dir" is not
input_files = args.inputs
datatype = args.datatype
dataset = args.dataset
pressure = args.pressure
quality_flag = args.use_quality_flag
lat_0 = args.lat_0
lon_0 = args.lon_0
bounding_lat = args.bounding_lat
extent = args.extent
plot_type = args.plot_type
#!/usr/bin/env python
encoding: utf-8
"""
thermo.py
Various thermodynamic relationship for ice and water.
Created by Geoff Cureton <geoff.cureton@ssec.wisc.edu> on 2009-04-04.
Copyright 2009-2025 University of Wisconsin Regents. All rights reserved.
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.
You should have received a copy of the GNU General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.
"""
# from scipy import log10
from numpy import log10
from numpy import exp, log
from numpy import min, minimum, where
# --
# Here we go. A set of functions that I use from time to time to calculate
# the basic stuff that I'm sick of doing over and over! I'm going to
# endeavour to include references and global constants to make it all nice
# and legible.
# --
Rs_da = 287.05 # Specific gas const for dry air, J kg^{-1} K^{-1}
Rs_v = 461.51 # Specific gas const for water vapour, J kg^{-1} K^{-1}
Cp_da = 1004.6 # Specific heat at constant pressure for dry air
Cv_da = 719.0 # Specific heat at constant volume for dry air
Cp_v = 1870.0 # Specific heat at constant pressure for water vapour
Cv_v = 1410.0 # Specific heat at constant volume for water vapour
Cp_lw = 4218 # Specific heat at constant pressure for liquid water
Epsilon = 0.622 # Epsilon=Rs_da/Rs_v; The ratio of the gas constants
degCtoK = 273.15 # Temperature offset between K and C (deg C)
rho_w = 1000.0 # Liquid Water density kg m^{-3}
grav = 9.81 # Gravity, m s^{-2}
Lv = 2.5e6 # Latent Heat of vaporisation
boltzmann = 5.67e-8 # Stefan-Boltzmann constant
mv = 18.0153 # Mean molar mass of water vapor(g/mol)
<v/git/cspp_sounder_ql-worktrees/legacy
NORMAL legacy scripts/sounder_image.py ^Wh 23 25% 49:9 19:35
thermo.py 9:18
leo
0 Home 1 Root 2 package 3 Repo* 4 Work- 5 pkg_test 6 python 7 SampleData
2026-01-08 19:35:06
1: geoffc@subrahmanyan:~/Computer_Stuff/Python/MadPy/Your
2: geoffc@subrahmanyan:~/SSEC/CSPP/CSPP-ActiveFire/git/cs
3: geoffc@subrahmanyan:~

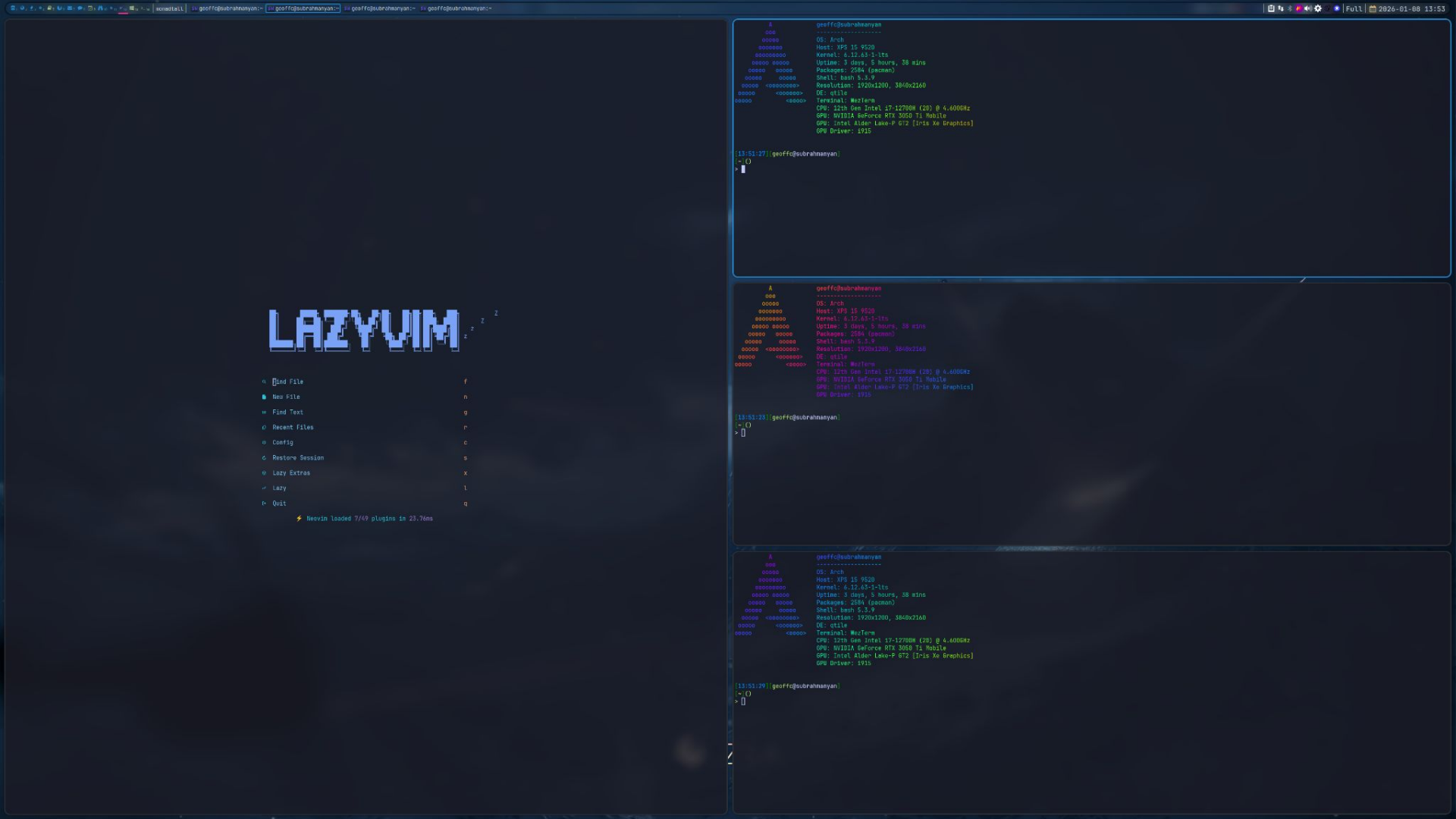
Tiling Window Managers

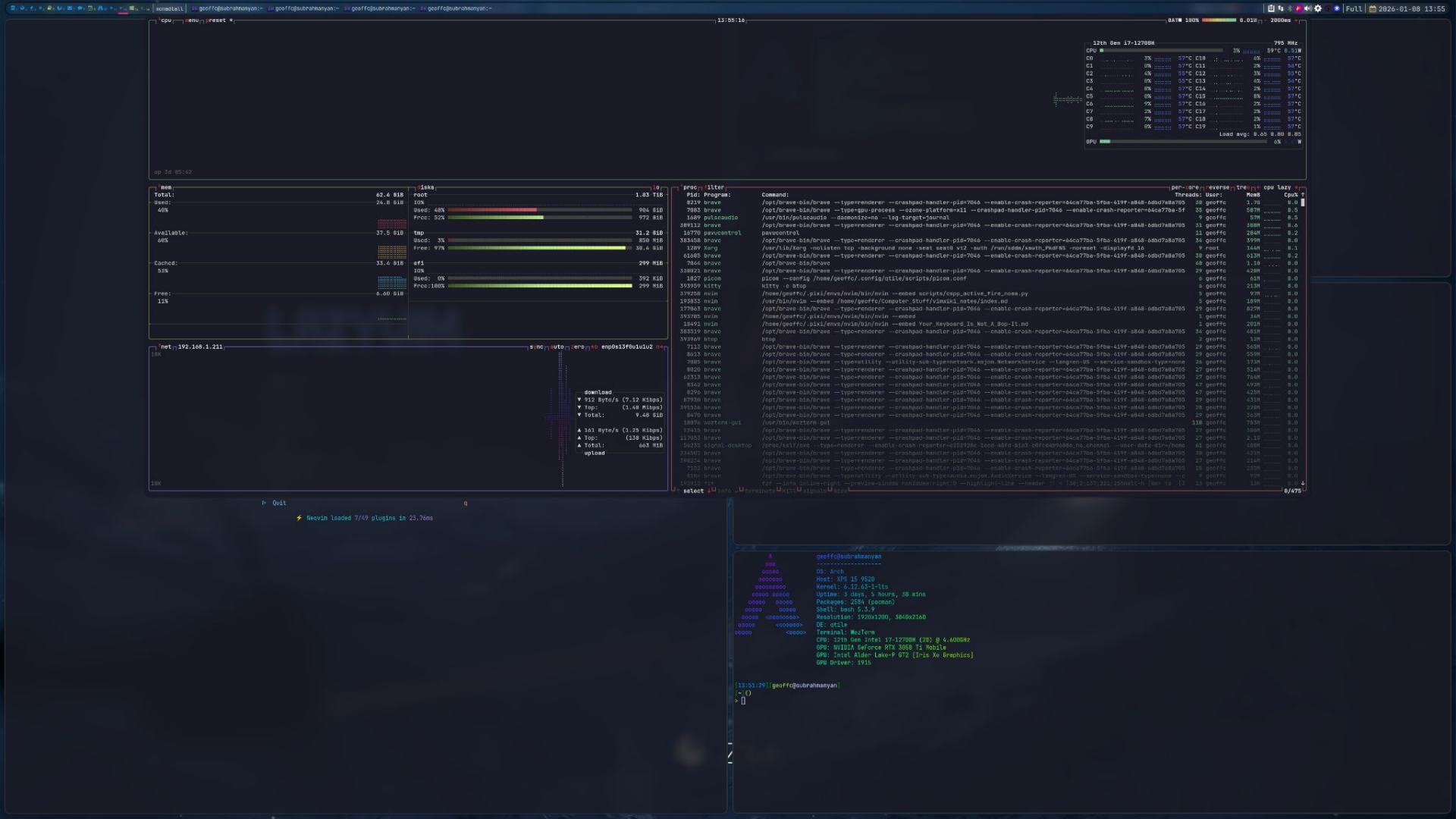
- Outside of my text editor (NeoVim), I try to manage my various desktop tasks in the same way as my editor: using the keyboard.
- The long-standing mode for app management is app-per-window, so how do I manage my windows?
- My preferred method is to use a tiling window manager, which manages where and how windows are drawn on the screen, so that the windows don't overlap.
- This allows the user to efficiently move windows around using screen, and between virtual desktops, using the keyboard.
- My preferred TWM is Qtile, written in python (https://pawamoy.github.io/qtile/)

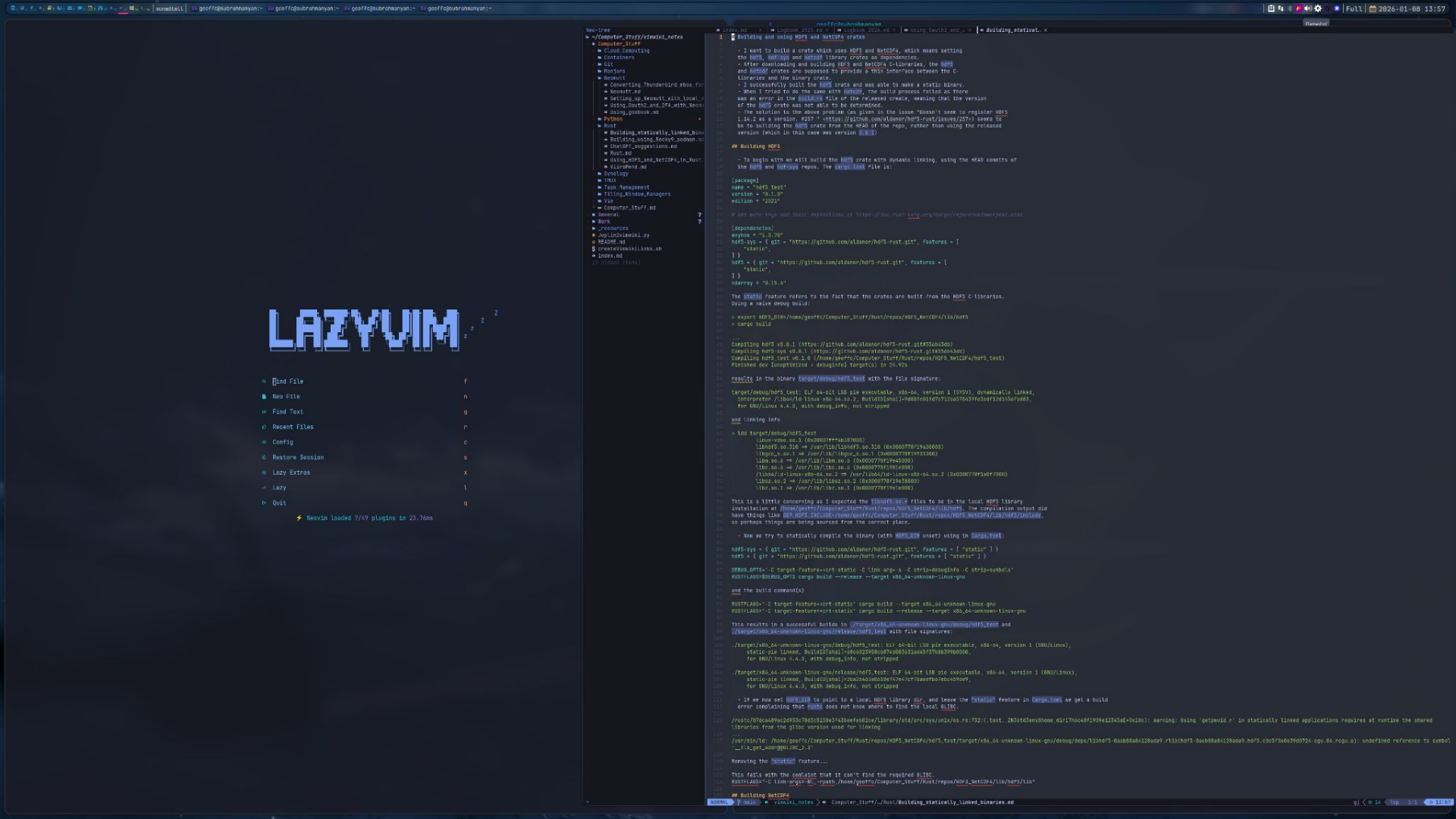
ZSA


monadtall
Full
2026-01-08 13:48
1
2
3
4
5
6
7
8
9
10
11
12
13
14
monadtall
ZSA


nomadtall
geoffc@subrahmanyan:~
geoffc@subrahmanyan:~
geoffc@subrahmanyan:~
Full
2026-01-08 13:52
geoffc@subrahmanyan
OS: Arch
Host: XPS 15 9520
Kernel: 6.12.63-1-lts
Uptime: 3 days, 5 hours, 38 mins
Packages: 2584 (pacman)
Shell: bash 5.3.9
Resolution: 1920x1200, 3840x2160
DE: qtile
Terminal: WezTerm
CPU: 12th Gen Intel i7-12700H (20) @ 4.600GHz
GPU: NVIDIA GeForce RTX 3050 Ti Mobile
GPU: Intel Alder Lake-P GT2 [Iris Xe Graphics]
GPU Driver: i915
[13:51:23][geoffc@subrahmanyan]
~
LAZYVIM
Find File
New File
Find Text
Recent Files
Config
Restore Session
Lazy Extras
Lazy
Quit
Neovim loaded 7/49 plugins in 23.76ms
[13:51:27][geoffc@subrahmanyan]
[13:51:29][geoffc@subrahmanyan]


LAZYVIM
Find File f
New File n
Find Text g
Recent Files r
Config c
Restore Session s
Lazy Extras x
Lazy l
Quit q
⚡ Neovim loaded 7/49 plugins in 23.76ms
geoffc@subrahmanyan
OS: Arch
Host: XPS 15 9520
Kernel: 6.12.63-1-lts
Uptime: 3 days, 5 hours, 38 mins
Packages: 2584 (pacman)
Shell: bash 5.3.9
Resolution: 1920x1200, 3840x2160
DE: qtile
Terminal: WezTerm
CPU: 12th Gen Intel i7-12700H (20) @ 4.600GHz
GPU: NVIDIA GeForce RTX 3050 Ti Mobile
GPU: Intel Alder Lake-P GT2 [Iris Xe Graphics]
GPU Driver: i915
[13:51:27][geoffc@subrahmanyan]
[13:51:23][geoffc@subrahmanyan]
[13:51:29][geoffc@subrahmanyan]


geoffc@subrahmanyan:~
geoffc@subrahmanyan:~
geoffc@subrahmanyan:~
geoffc@subrahmanyan:~
Full
2026-01-08 13:55
cpu
menu
reset
13:55:16
BAT 100%
8.01W
2800ms
12th Gen i7-12700H
795 MHz
CPU
3%
59°C
6.51W
C0
C1
C2
C3
C4
C5
C6
C7
C8
C9
GPU
Load avg: 8.05 8.80 8.85
up 3d 05:42
mem
Total:
62.4 GiB
Used:
24.8 GiB
40%
Available:
37.5 GiB
60%
Cached:
33.4 GiB
53%
Free:
6.40 GiB
11%
disks
io
root
1.83 TiB
IO%
Used: 48%
904 GiB
Free: 52%
972 GiB
tmp
31.2 MiB
Used: 3%
850 KiB
Free: 97%
30.4 GiB
ef1
299 MiB
IO%
Used: 0%
392 KiB
Free:100%
299 MiB
net
192.168.1.211
sync
auto
zero
enp0s13f0u1u1u2
download
912 Byte/s (7.12 Kibps)
Top:
(3.48 Mibps)
Total:
9.48 GiB
upload
161 Byte/s (1.25 Kibps)
Top:
(138 Kibps)
Total:
603 MiB
proc
filter
per-core
reverse
tree
cpu lazy
Pid:
Program:
Command:
Threads:
User:
Mem%
Cpu%
8219 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 38 geoffc 1.7G 8.0
7083 brave /opt/brave-bin/brave --type=gpu-process --ozone-platform=x11 --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5f 35 geoffc 587M 8.5
1689 pulseaudio /usr/bin/pulseaudio --daemonize=no --log-target=journal 9 geoffc 57M 8.5
389112 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 33 geoffc 388M 8.0
16779 pavucontrol pavucontrol 11 geoffc 264M 8.2
383458 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 34 geoffc 399M 8.0
1289 Xorg /usr/lib/Xorg -nolisten tcp -background none -seat seat0 vt2 -auth /run/sddm/xauth_PkdFNS -noreset -displayfd 16 9 root 144M 8.1
61685 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 30 geoffc 613M 8.2
7846 brave /opt/brave-bin/brave 48 geoffc 1.1G 8.0
338021 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 428M 8.0
1827 picom picom --config /home/geoffc/.config/qtile/scripts/picom.conf 6 geoffc 61M 8.0
393959 kitty kitty -o btop 6 geoffc 213M 8.0
379258 nvim /home/geoffc/.pixi/envs/nvim/bin/nvim --embed scripts/cspp_active_fire_noaa.py 5 geoffc 97M 8.0
193033 nvim /usr/bin/nvim --embed /home/geoffc/Computer_Stuff/vimwiki_mates/index.md 5 geoffc 109M 8.0
177863 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 827M 8.0
393785 nvim /home/geoffc/.pixi/envs/nvim/bin/nvim --embed 1 geoffc 34M 8.0
18491 nvim /home/geoffc/.pixi/envs/nvim/bin/nvim --embed Your_Keyboard_Is_Not_A_Bop-It.nd 1 geoffc 201M 8.0
383519 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 34 geoffc 481M 8.0
393969 btop btop 3 geoffc 12M 8.0
7113 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 565M 8.0
8613 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 559M 8.0
7085 brave /opt/brave-bin/brave --type=utility --utility-sub-type=network.mojom.NetworkService --lang=en-US --service-sandbox-type=none 26 geoffc 173M 8.0
8020 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 27 geoffc 514M 8.0
62313 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 27 geoffc 764M 8.0
8342 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 47 geoffc 493M 8.0
8296 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 47 geoffc 425M 8.0
87936 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 431M 8.0
39153s brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 28 geoffc 239M 8.0
8479 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 29 geoffc 363M 8.0
18876 wezterm-gui /usr/bin/wezterm-gui 110 geoffc 783M 8.0
12415 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 27 geoffc 386M 8.0
117853 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 27 geoffc 2.1G 8.0
36233 signal-desktop /proc/self/exe --type=renderer --enable-crash-reporter=e252928c-1ced-4bf3-b1c3-c8fc0a9688ac,no_channel --user-data-dir=/home 61 geoffc 468M 8.0
234853 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 38 geoffc 421M 8.0
59234 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 27 geoffc 214M 8.0
7132 brave /opt/brave-bin/brave --type=renderer --crashpad-handler-pid=7046 --enable-crash-reporter=a4ca77ba-5fba-419f-a848-ddbd7a8a705 26 geoffc 235M 8.0
8186 brave /opt/brave-bin/brave --type=utility --utility-sub-type=audio.mojom.AudioService --lang=en-US --service-sandbox-type=none --s 9 geoffc 42M 8.0
193913 fzf fzf --info inline-right --preview-window nohidden:right:0 --highlight-line --header ' ^ (38;2;137;221;255mAlt-h [0m↑ to [3 15 geoffc 13M 8.0
select
terminate
kill
signals
6/475
Quit
q
Neovim loaded 7/49 plugins in 23.76ms
geoffc@subrahmanyan
OS: Arch
Host: XPS 15 9520
Kernel: 6.12.63-1-lts
Uptime: 3 days, 5 hours, 38 mins
Packages: 2584 (pacman)
Shell: bash 5.3.9
Resolution: 1920x1200, 3840x2160
DE: qtile
Terminal: WezTerm
CPU: 12th Gen Intel i7-12700H (20) @ 4.600GHz
GPU: NVIDIA GeForce RTX 3050 Ti Mobile
GPU: Intel Alder Lake-P GT2 [Iris Xe Graphics]
GPU Driver: i915
[13:51:29][geoffc@subrahmanyan]
~ ()
>
/


LAZYVIM
Find File f
New File n
Find Text g
Recent Files r
Config c
Restore Session s
Lazy Extras x
Lazy l
Quit q
Neovim loaded 7/49 plugins in 25.76ms

Thank you!