Saul Dotfiles

This repo contains some of my config files for ArchLinux on Acer Aspire 5. The files in root/ are symlinked to the correct location in my system. I am using this repo mainly as a backup and documentation of my configuration. `fstab` does not work when symlinked so it must be copied and kept track of manually. The folder installed_packages is an update list of ll the packages present on my system.

How To Use

- My scripts are available on Github at `saulpierotti/.scripts`
- Use `to_dotfiles.sh filename` to copy a file to this repo and replace it with a symlink in the original location
- For consistency, I always symlink files and never folders
- The script `restore_dotfiles.sh` takes all the files in the folder and replaces the original file locations with symlinks to them
  - ATTENTION: this overwrites the files in the system

TODO

Notes to Myself

Bootloader

- I am using Grub2 with the `arch-silence-grub-theme-git` theme
- I use `grub-btrfs` to see also btrfs snapshots in the boot menu

Kernel

- I use linux-zen and I also installed its headers
- I removed the vanilla kernel to avoid 2 lengthy compilations at every package update

GPU

- For managing the Nvidia and Intel GPUs, I am using Optimus Manager
  - Switching between GPUs and to hybrid mode works
  - In Nvidia mode TTY changes are not recommended and can result in freezes (but seems to work using the `xf86-video-intel` driver and setting intel mode instead of modesetting in optimus-manager-qt)
  - When resuming from a lock screen in nvidia or hybrid mode, the script Xorg post-start hook `/sbin/prime-offload` is not called
    - Call it manually when needed (`prime-offload`) or reload i3 (I put prime_offload_hack.sh as an `exec_always`)
    - It is needed when I see a really long string in conky
- For Nvidia I use the `nvidia-dkms` driver (since I am using the linux-zen kernel)

Dual monitor and resolution

- In general monitors and resolutions are managed by xrandr
- It is easier to use arandr

Lock Screen and Screensaver

- As a screen-locker I am using light-locker, which just redirects to the LightDM login page
  - Light locker 1.9.0.3 has a bug that causes in a blank screen after unlock, reported in the Arch package page
  - 1.9.0.2 works fine, and thus I added light-locker to the list of packages ignored by pacman in `/etc/pacman.conf`
- The locker is called when xset (part of xorg) activates the screensaver

  - The screensaver activation behavior is managed permanently in `/etc/X11/xorg.conf.d/10-monitor.conf`
  - I can inspect the current settings with `xset q`
- I can invoke the lock manually by calling `light-locker-command -l`
- The command light-locker is a background process that needs to be in execution for the lock to work
  - I execute it in the i3 config file
  - The execution parameters of the command can tweak the behavior of the lock (when to lock after the X screensaver is activated)
- For some strange reason, the login after lock is done in TTY8, while LightDM is on TTY7
- Given the problem for TTY switching with Nvidia, do not use the lock when in Nvidia mode
  - If the screen gets locked in nvidia mode: press space, then write the passwrd and press enter. The lockscreen is working even if the screen no
  - After unlocking like this once, then the screen lock works in nvidia mode until the next logout
- Inhibition of the lockscreen is done with exam_mode.sh and inhibited with exam_mode_undo.sh
  - They just call xset and disable/enable blanking and screen poweroff

Login Manager

- I am using LightDM
- If stuck in a login loop (password is accepted but then LighDM comes again)
  - Usually there is some error in .profile, .zshrc, .zshenv, or similar file read at login
  - If not, it can be a permission problem on such files
  - If not, it can be a permission problem on the folder /tmp

LightDM

- I am using the webkit2 `lightdm-webkit-theme-litarvan` theme
- The user avatar and name (Saul Pierotti, not the user name) are better managed from the gnome settings panel (login to gnome!)
  - For some reason when I change the picture manually it does not work

Window Manager

- I am using i3-gapps, and I keep gnome as a fallback
- In i3, I use polybar as a status bar
- Notifications are managed by twmnc
- The wallpaper is set by feh in `~/.xsession`
  - Note: it has to have executable permissions!

File Browser

- I use ranger mostly, and nautilus occasionally when I need to do something that is easier graphicly

Ranger

- Ranger can be started from the command line or with its .desktop file
- When started from the command line, it heredits the environment of the user shell (.zshrc, .zsh_aliases, .zshenv)
- When called as a desktop application, it does not heredit the shell environment of the user
  - This can be a source of problems mainly for applications called by ranger that need conda
  - When opening a python script from ranger, this is opened in nvim, and since condainit is not sourced it is not possible to change conda environment from within vim
- For these conda problems, I use this setup
  - I created a custom desktop file in `.local/share/applications/ranger.desktop` which has precedence over the one in `/usr/share/applications`
  - This file does not execute ranger but it executes a script that I placed in `~/.scripts/ranger_visual_launcher.`
  - This script sources `~/.condainit` and then executes ranger on the folder `$RANGER_START` (which is set from `.zshenv`)

  - I do not source directly `~/.zshrc` since it cause lags when opening ranger (there are all the Oh-My-Zsh plugins)
  - Now it is possible to change conda environment from ranger launched as a desktop app in vim using `:CondaChangeEnv`

Text Editor

- I use nvim

Nvim

- I disabled the background color so to use the terminal background with `highlight Normal ctermbg=none`
  - This must be after the theme declaration to be effective
- I use the plugin `cjrh/vim-conda` for changing conda environments inside vim
  - This plugin requires condainit to be sourced in the shell that calls nvim
  - When calling vim from the command line this is not a problem
  - When opening vim from ranger (just opening a file) .zshrc is not sourced and so it does not work
    * This is a problem only if ranger is started as a desktop app and not from command line
    * To overcome this, I source condainit before opening ranger as a desktop app (see ranger section)
- I use the ALE plugin as a linter and fixer
  - Conda environments in ALE can be changed from within vim with vim-conda (`:CondaChangeEnv`)
  - I created the shortcut `ca` to change environment in that way and then call a script that configures the envirionment also for pyright (see Coc.nvim)
- I use Coc.nvim for autocompletion
  - Since I already use ALE for linting, I set up CoC to redirect linting messages to ALE in its config file (of Coc)
  - Coc can be configured at `.config/nvim/coc-settings.json`
  - Coc plugins are used for enabling intellinsense in different languages
  - Extensions can be installed with `:CocInstall`, but I prefer using Plug on my `init.nvim`
    * In order to install Coc extensions with Plug you need `yarn` (install with pacman)
    * Add the post-update hook as `Plug 'neoclide/coc-json', {'do': 'yarn install --forzen-lockfile'}`
    * Extensions installed with Plug cannt be removed with `:CocUnsinstall`
  - For python I use coc-pyright (Microsoft LSP)
    * It has problems with conda environments, and it does not see the env set by vim-conda
    * It does not also recognise configuration files in the root of the project folder
    * The only working way to set the path is to change `python.pythonPath` in the Coc configuration file
    * I change its envirionment using the shortcut `ca`, that first calls vim-conda (so to set the env for ALE)
      · After setting an env with vim-conda, it calls a script that overwrites the Coc config file
      · It sets `python.pythonPath` to what is selected by vim-conda
  - In general I am disabling the linting features of Coc since I prefer ALE for that
- I use `puremourning/vimspector` for debugging
- It is better to avoid using vim within a conda environment for problems with vim-conda and Coc.nvim

Terminal

- I use kitty since it supports fonts with ligatures (I use Fira Code) and it is GPU-accellerated

Kitty

- I did not set zsh autocompletion for kitty in `.zshrc` since I am using zsh-completions with antigen
- Since the `$TERM` definition `xterm-kitty` is not common, in ssh often you get a prompt without color

  - To overcome this, I aliased `ssh` to `TERM='xterm-256color' ssh $@` in `~/.zsh_aliases`

Console

- I use zsh and leave bash and its basic config files for compatibility

zsh

- I use antigen as a plugin manager
- I use the oh-my-zsh plugin library

Connectivity

- I use network-manager, managed mostly via the nm-applet
- I implemented a WiFi hotspot with network-manager and dnsmasq
  - For connection sharing, netowrk manager requires the optional dependency `dnsmasq`
  - I created a fake WiFi network from network manager using WPA2 encription and a key longer than 8 charachters
  - The mode of the network must be set to Hotspot
  - To activate the hotspot, just connect to the network (connect to hidden network and select hotspot, or from command line)
- I set `dnsmasq` as the default dns method for NetworkManager in `/etc/NetwrokManager/conf.d/00-use-dnsmasq.co`
- For the biocomp VPN to work, you have to go to the IPV4 tab in the VPN page of `nm-connection-editor`, go to Routes and select "Use this connection only for resources in this network"

Media

- I use mpv for videos with some configuration to improve quality at a performance cost (in `~/.config/mpv/mpv.conf`)

NordVPN

- Do not sync the config file since it contains an access token

Tex

- I installed the texlive-most meta package
- I edit in vim using the plugin vimtex, and ALE for linting
- ALE uses latexindent, which is included in texlive-most, but misses the perl module `perl-log-dispatch` (in the AUR)
- I use coc-texlab for completion