

Dotfiles

- [Sway Config](#)
- [Automatic Cloudflare DNS Update](#)
- [NixOS Config](#)
 - [configuration.nix](#)
 - [bemenu.nix](#)
 - [hypridle.nix](#)
 - [hyprland.nix](#)
 - [hyprlock.nix](#)
 - [kitty.nix](#)
 - [neovim.nix](#)
 - [waybar.nix](#)
 - [zsh.nix](#)
 - [tmux.conf](#)
 - [nixvim.nix](#)
 - [shared-home.nix](#)

Sway Config

This will be semi-automatically updated via the [Bookstack REST API](#)

Sway Config

```
# Default config for sway
#
# Copy this to ~/.config/sway/config and edit it to your liking.
#
# Read `man 5 sway` for a complete reference.

### Variables
#
# Logo key. Use Mod1 for Alt.
set $mod Mod4
# Home row direction keys, like vim
set $left h
set $down j
set $up k
set $right l
# Your preferred terminal emulator
#set $term kitty
set $term foot -f "Noto Sans Mono:size=12"
# Your preferred application launcher
# Note: pass the final command to swaymsg so that the resulting window can be opened
# on the original workspace that the command was run on.
#set $menu dmenu_path | dmenu | xargs swaymsg exec --
set $menu bemenu-run --fn "Source Code Pro" 20 | xargs swaymsg exec --

### Output configuration
#
# Default wallpaper (more resolutions are available in /usr/share/backgrounds/sway/)
#output * bg /usr/share/backgrounds/sway/Sway_Wallpaper_Blue_1920x1080.png fill
#
```

```

# Example configuration:
#
# output HDMI-A-1 resolution 1920x1080 position 1920,0
#
# You can get the names of your outputs by running: swaymsg -t get_outputs

### Idle configuration
#
# Example configuration:
#
# exec swayidle -w \
#     timeout 300 'swaylock -f -c 000000' \
#     timeout 600 'swaymsg "output * power off"' resume 'swaymsg "output * power on"' \
#     before-sleep 'swaylock -f -c 000000'
#
# This will lock your screen after 300 seconds of inactivity, then turn off
# your displays after another 300 seconds, and turn your screens back on when
# resumed. It will also lock your screen before your computer goes to sleep.

### Input configuration
#
# Example configuration:
#
# input "2:14:SynPS/2_Synaptics_TouchPad" {
#     dwt enabled
#     tap enabled
#     natural_scroll enabled
#     middle_emulation enabled
# }
#
# You can get the names of your inputs by running: swaymsg -t get_inputs
# Read `man 5 sway-input` for more information about this section.

input type:touchpad {
    tap enabled
    natural_scroll enabled
    accel_profile "flat"
}

```

Key bindings

#

Basics:

#

Start a terminal

bindsym \$mod+Return exec \$term

Kill focused window

bindsym \$mod+q kill

Start your launcher

bindsym \$mod+Shift+d exec \$menu

Drag floating windows by holding down \$mod and left mouse button.

Resize them with right mouse button + \$mod.

Despite the name, also works for non-floating windows.

Change normal to inverse to use left mouse button for resizing and right

mouse button for dragging.

floating_modifier \$mod normal

Reload the configuration file

bindsym \$mod+Shift+c reload

Exit sway (logs you out of your Wayland session)

bindsym \$mod+Shift+e exec swaymsg -t warning -m 'You pressed the exit shortcut. Do you really want to exit sway? This will end your Wayland session.' -B 'Yes, exit sway' 'swaymsg exit'

#

Moving around:

#

Move your focus around

bindsym \$mod+\$left focus left

bindsym \$mod+\$down focus down

bindsym \$mod+\$up focus up

bindsym \$mod+\$right focus right

Or use \$mod+[up|down|left|right]

bindsym \$mod+Left focus left

bindsym \$mod+Down focus down

bindsym \$mod+Up focus up

bindsym \$mod+Right focus right

Move the focused window with the same, but add Shift

bindsym \$mod+Shift+\$left move left

bindsym \$mod+Shift+\$down move down

bindsym \$mod+Shift+\$up move up

bindsym \$mod+Shift+\$right move right

Ditto, with arrow keys

bindsym \$mod+Shift+Left move left

bindsym \$mod+Shift+Down move down

bindsym \$mod+Shift+Up move up

bindsym \$mod+Shift+Right move right

#

Workspaces:

#

Switch to workspace

bindsym \$mod+1 workspace number 1

bindsym \$mod+2 workspace number 2

bindsym \$mod+3 workspace number 3

bindsym \$mod+4 workspace number 4

bindsym \$mod+5 workspace number 5

bindsym \$mod+6 workspace number 6

bindsym \$mod+7 workspace number 7

bindsym \$mod+8 workspace number 8

bindsym \$mod+9 workspace number 9

bindsym \$mod+0 workspace number 10

Move focused container to workspace

bindsym \$mod+Shift+1 move container to workspace number 1, workspace number 1

bindsym \$mod+Shift+2 move container to workspace number 2, workspace number 2

bindsym \$mod+Shift+3 move container to workspace number 3, workspace number 3

bindsym \$mod+Shift+4 move container to workspace number 4, workspace number 4

bindsym \$mod+Shift+5 move container to workspace number 5, workspace number 5

bindsym \$mod+Shift+6 move container to workspace number 6, workspace number 6

bindsym \$mod+Shift+7 move container to workspace number 7, workspace number 7

bindsym \$mod+Shift+8 move container to workspace number 8, workspace number 8

bindsym \$mod+Shift+9 move container to workspace number 9, workspace number 9

bindsym \$mod+Shift+0 move container to workspace number 10, workspace number 10

Note: workspace numbers can have any name you want, not just numbers.

We just use 1-10 as the default.

```
# Set monitor variables
set $internal_monitor eDP-1
set $second_monitor DP-1

# Assign workspaces to separate monitors

output $internal_monitor pos 0 0
    output $second_monitor pos 1080 0

workspace 1 output $internal_monitor
workspace 2 output $internal_monitor
workspace 3 output $internal_monitor
workspace 4 output $internal_monitor
workspace 5 output $internal_monitor
workspace 6 output $second_monitor
workspace 7 output $second_monitor
workspace 8 output $second_monitor
workspace 9 output $second_monitor
workspace 10 output $second_monitor

#
# Layout stuff:
#
# You can "split" the current object of your focus with
# $mod+b or $mod+v, for horizontal and vertical splits
# respectively.
bindsym $mod+n splith
bindsym $mod+v splitv

# Switch the current container between different layout styles
bindsym $mod+s layout stacking
bindsym $mod+z layout tabbed
bindsym $mod+e layout toggle split

# Make the current focus fullscreen
bindsym $mod+f fullscreen

# Toggle the current focus between tiling and floating mode
bindsym $mod+Shift+space floating toggle
```

```
# Swap focus between the tiling area and the floating area
bindsym $mod+space focus mode_toggle

# Move focus to the parent container
bindsym $mod+a focus parent

#
# Scratchpad:
#
# Sway has a "scratchpad", which is a bag of holding for windows.
# You can send windows there and get them back later.

# Move the currently focused window to the scratchpad
bindsym $mod+Shift+minus move scratchpad

# Show the next scratchpad window or hide the focused scratchpad window.
# If there are multiple scratchpad windows, this command cycles through them.
bindsym $mod+minus scratchpad show

#
# Resizing containers:
#
mode "resize" {
    # left will shrink the containers width
    # right will grow the containers width
    # up will shrink the containers height
    # down will grow the containers height
    bindsym $left resize shrink width 100px
    bindsym $down resize grow height 100px
    bindsym $up resize shrink height 100px
    bindsym $right resize grow width 100px

    # Ditto, with arrow keys
    bindsym Left resize shrink width 100px
    bindsym Down resize grow height 100px
    bindsym Up resize shrink height 100px
    bindsym Right resize grow width 100px

    # Return to default mode
    bindsym Return mode "default"
```

```
bindsym Escape mode "default"
}

bindsym $mod+r mode "resize"

#
# Status Bar:
#
# Read `man 5 sway-bar` for more information about this section.
bar {
    position top

    # When the status_command prints a new line to stdout, swaybar updates.
    # The default just shows the current date and time.
    #status_command while date +%Y-%m-%d %l:%M:%S %p'; do sleep 1; done
    status_command while ~/.config/sway/status.sh; do sleep 1; done

    colors {
        statusline #ffffff
        background #323232aa
        inactive_workspace #323232aa #323232aa #000000
    }
}

[]
[]font "Source Code Pro"
[]pango_markup enabled
[]separator_symbol "$"
}

#
# Border Controls
#

default_border pixel
bindsym $mod+Shift+b exec swaymsg border toggle

#
# Startup Applications
#

exec swaybg -i $(find ~/Pictures/wallpapers -type f | shuf -n1) -m fill &
```

#

Power Menu

#

bindsym \$mod+x mode "(k) lock, (l) logout, (r) reboot, (s) shutdown"

mode "(k) lock, (l) logout, (r) reboot, (s) shutdown" {

▢bindsym k exec swaylock -c 000000, mode "default"

▢bindsym l exec swaymsg exit, mode "default"

▢bindsym r exec reboot, mode "default"

▢bindsym s exec shutdown now, mode "default"

▢bindsym Return mode "default"

▢bindsym Escape mode "default"

}

#

Special Keybindings

#

Map Caps Lock to Left Control

input type:keyboard {

▢xkb_options caps:ctrl_modifier

}

Applications

bindsym \$mod+w exec flatpak run io.gitlab.librewolf-community

Volume

bindsym XF86AudioRaiseVolume exec pactl set-sink-volume @DEFAULT_SINK@ +5% && pactl set-sink-mute @DEFAULT_SINK@ 0

bindsym XF86AudioLowerVolume exec pactl set-sink-volume @DEFAULT_SINK@ -5% && pactl set-sink-mute @DEFAULT_SINK@ 0

bindsym XF86AudioMute exec pactl set-sink-mute @DEFAULT_SINK@ toggle

bindsym XF86AudioMicMute exec pactl set-source-mute @DEFAULT_SOURCE@ toggle

Media

bindsym XF86AudioPlay exec playerctl play-pause

bindsym XF86AudioNext exec playerctl next

bindsym XF86AudioPrev exec playerctl previous

```
bindsym XF86Tools exec playerctl play-pause
bindsym XF86Favorites exec playerctl next
bindsym XF86Bluetooth exec playerctl previous
```

#Sticky Window

```
bindsym $mod+Shift+Return sticky toggle
```

Brightness

```
bindsym XF86MonBrightnessUp exec brightnessctl set +10%
bindsym XF86MonBrightnessDown exec brightnessctl set --min-value=1 10%-
```

Rotation

```
exec swaymsg input 1386:20824:Wacom_Pen_and_multitouch_sensor_Finger map_to_output
$internal_monitor
bindsym $mod+Mod1+k exec swaymsg output eDP-1 transform normal && swaymsg input
1386:20824:Wacom_Pen_and_multitouch_sensor_Finger map_to_output $internal_monitor && swaymsg
input 1386:20824:Wacom_Pen_and_multitouch_sensor_Pen map_to_output $internal_monitor
bindsym $mod+Mod1+j exec swaymsg output eDP-1 transform 180 && swaymsg input
1386:20824:Wacom_Pen_and_multitouch_sensor_Finger map_to_output $internal_monitor && swaymsg
input 1386:20824:Wacom_Pen_and_multitouch_sensor_Pen map_to_output $internal_monitor
bindsym $mod+Mod1+l exec swaymsg output eDP-1 transform 90 && swaymsg input
1386:20824:Wacom_Pen_and_multitouch_sensor_Finger map_to_output $internal_monitor && swaymsg
input 1386:20824:Wacom_Pen_and_multitouch_sensor_Pen map_to_output $internal_monitor
bindsym $mod+Mod1+h exec swaymsg output eDP-1 transform 270 && swaymsg input
1386:20824:Wacom_Pen_and_multitouch_sensor_Finger map_to_output $internal_monitor && swaymsg
input 1386:20824:Wacom_Pen_and_multitouch_sensor_Pen map_to_output $internal_monitor

include /etc/sway/config.d/*
```

status.sh

```
date=$(date +%Y-%m-%d %l:%M:%S %p)
wifi_details=$(nmcli -t -f active,ssid,freq,rate,signal dev wifi list --rescan no | grep -e "^yes:" | sed
"s/yes://g;s:/ /g")
wired_info=$(ip addr | grep 'state UP' -A2 | tail -n1 | awk '{print $2}' | cut -f1 -d'/')
```

```

if [ -z "$wifi_details" ]; then
network_info="Wired $wired_info"
else
network_info="$wifi_details%"
fi

SINK=0
current_brightness=$(brightnessctl | grep Current | cut -d '(' -f2 | cut -d '%' -f1)
current_sink_muted=$(pactl list sinks | grep '^[[:space:]]Mute:' | head -n $(( $SINK + 1 )) | tail -n 1 | cut -d
":" -f2 | sed -e 's/^[ \t]*//')

if [ "$current_sink_muted" = "yes" ]; then
current_volume="muted"
else
current_volume=$(pactl list sinks | grep '^[[:space:]]Volume:' | head -n $(( $SINK + 1 )) | tail -n 1 | sed -e
's,.* \([0-9][0-9]*\)%.*,\1,')
fi

current_mem=$(free | grep Mem | awk '{print $3/$2 * 100.0}')
current_mem_rounded=$(printf "%.2f" $current_mem)
current_cpu=$(100-$(vmstat 1 2|tail -1|awk '{print $15}'))
current_cpu_rounded=$(printf "%.2f" $current_cpu)
battery_level=$(cat /sys/class/power_supply/BAT0/capacity)
battery_status=$(cat /sys/class/power_supply/BAT0/status)
if [ "$battery_status" = "Charging" ]; then
battery_prefix="+"
fi

if [ "$battery_level" -le "15" ]; then
battery_color="red"
else
battery_color="springgreen"
fi

echo "<span foreground='limegreen' weight='bold'>$network_info</span> <span foreground='palegreen'
weight='bold'>CPU:$current_cpu%</span> <span foreground='mediumspringgreen'
weight='bold'>MEM:$current_mem_rounded</span> <span foreground='$battery_color'
weight='bold'>BATT:$battery_prefix$battery_level</span> <span foreground='mediumseagreen'
weight='bold'>LUM:$current_brightness</span> <span foreground='yellowgreen'
weight='bold'>VOL:$current_volume</span> <span foreground='lightseagreen'

```

weight='bold'>\$date "

Automatic Cloudflare DNS Update

Create a script that queries the current IP address, then updates it using Cloudflare's API

```
# Acquire the permanent, publically addressable ipv6 address of the eth0 adapter
ip -6 addr show dev eth0 mngtmpaddr | grep -oE "([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}"
```

```
# Acquire the ipv4 address
curl https://api.ipify.org
```

```
import http.client requests json
```

```
ipv4_data = json.loads(requests.get('https://api.ipify.org?format=json').text)
```

```
ipv4_addr = ipv4_data["ip"]
```

```
# Acquire the permanent, publically addressable ipv6 address of the eth0 adapter
```

```
ipv6_addr = subprocess.run('ip -6 addr show dev eth0 mngtmpaddr | grep -oE "([0-9a-fA-F]{1,4}:){7}[0-9a-fA-F]{1,4}"')
```

```
# Acquire zones
```

```
conn = http.client.HTTPSConnection("api.cloudflare.com")
```

```
headers = {
```

```
    'Content-Type': "application/json",
```

```
    'Authorization': "Bearer <API_TOKEN>"
```

```
}
```

```
conn.request("GET", "/client/v4/zones", headers=headers)
```

```
res = conn.getresponse()
```

```
data = res.read()
```

```
# Modify all DNS records
```

```
...
```


NixOS Config

configuration.nix

```
{ config, lib, pkgs, ... }:  
  
{  
  imports =  
    [ ./hardware-configuration.nix ];  
  
  # Use the systemd-boot EFI boot loader.  
  boot.loader.systemd-boot.enable = true;  
  boot.loader.efi.canTouchEfiVariables = true;  
  
  nix.settings.experimental-features = [ "nix-command" "flakes" ];  
  nix.optimise.automatic = true;  
  nixpkgs.config.allowUnfree = false;  
  
  networking.hostName = "coracle";  
  networking.networkmanager.enable = true;  
  
  hardware.bluetooth.enable = true;  
  hardware.bluetooth.powerOnBoot = false;  
  
  time.timeZone = "America/New_York";  
  
  fonts.packages = with pkgs; [  
    noto-fonts  
    noto-fonts-cjk  
    noto-fonts-emoji  
    liberation_ttf  
    fira-code  
    fira-code-symbols  
    nerdfonts  
  ];  
  
  security.rtkit.enable = true;
```

```
security.pam.services.hyprlock = {};
```

```
services.printing.enable = true;
```

```
services.udisks2.enable = true;
```

```
services.pipewire = {
```

```
    enable = true;
```

```
    alsa.enable = true;
```

```
    alsa.support32Bit = true;
```

```
    pulse.enable = true;
```

```
};
```

```
services.flatpak.enable = true;
```

```
xdg.portal = {
```

```
    enable = true;
```

```
    extraPortals = [ pkgs.xdg-desktop-portal-hyprland pkgs.xdg-desktop-portal-gtk ];
```

```
};
```

```
services.openssh = {
```

```
    enable = true;
```

```
    ports = [ 22 ];
```

```
    settings = {
```

```
        PasswordAuthentication = false;
```

```
        PermitRootLogin = "prohibit-password";
```

```
    };
```

```
};
```

```
services.auto-cpufreq.enable = true;
```

```
powerManagement.enable = true;
```

```
users.mutableUsers = true;
```

```
users.defaultUserShell = pkgs.zsh;
```

```
users.users.root = {
```

```
};
```

```
users.users.game = {
```

```
    isNormalUser = true;
```

```
    createHome = true;
```

```
    extraGroups = [ "networkmanager" ];
```

```
};
```

```
users.users.code = {
```

```
isNormalUser = true;
createHome = true;
extraGroups = [ "networkmanager" "adbusers" "dialout" ];
};
```

```
environment.systemPackages = [ ];
```

```
programs.zsh.enable = true;
programs.hyprland.enable = true;
programs.neovim = {
  enable = true;
  configure = {
    customRC = "
      set number relativenumber
      filetype plugin indent on
    ";
  };
  viAlias = true;
  vimAlias = true;
};
```

```
programs.tmux = {
  enable = true;
  extraConfig = import ./tmux.conf;
};
```

```
programs.firejail = {
  enable = true;
};
```

```
programs.adb.enable = true;
```

```
virtualisation.containers.enable = true;
```

```
virtualisation = {
  podman = {
    enable = true;
  };
};
```

```
networking.firewall.enable = false;
system.copySystemConfiguration = true;
system.autoUpgrade = {
  enable = true;
```

```
flake = "inputs.self.outPath";
flags = [
  "--update-input"
  "nixpkgs"
  "-L"
];
dates = "02:00";
randomizedDelaySec = "45min";
};

system.stateVersion = "24.11";

}
```

bemenu.nix

```
{ config, pkgs, lib, ... }:
```

```
{
  programs.bemenu = {
    enable = true;
    settings = {
      line-height = 28;
      fn = "Fira Code";
      fb = "#1e1e2e";
      ff = "#cdd6f4";
      nb = "#1e1e2e";
      nf = "#cdd6f4";
      tb = "#1e1e2e";
      hb = "#1e1e2e";
      tf = "#f38ba8";
      hf = "#f9e2af";
      af = "#cdd6f4";
      ab = "#1e1e2e";
    };
  };
}
```

hypridle.nix

```
{ config, pkgs, lib, ... }:

{
  services.hypridle = {
    enable = true;
    settings = {
      general = {
        before_sleep_cmd = "hyprlock --immediate";
        after_sleep_cmd = "hyprctl dispatch dpms on";
        ignore_dbus_inhibit = false;
        lock_cmd = "hyprlock --immediate";
      };
    };
  };
}
```

hyprland.nix

```
{ config, pkgs, lib, ... }:

{
  wayland.windowManager.hyprland = {
    enable = true;
    settings = {
      "$browser" = "flatpak run --user io.gitlab.librewolf-community";
      "$mainMod" = "SUPER";
      "$menu" = "bmenu-run --fn Fira Code";
      "$terminal" = "kitty";

      monitor=",highrr,auto,auto";
      exec-once="waybar";

      general = {
        allow_tearing=false;
        border_size=2;
        "col.active_border"="rgba(33ccffee) rgba(00ff99ee) 45deg";
        "col.inactive_border"="rgba(595959aa)";
        gaps_in=5;
        gaps_out=20;
        layout="dwindle";
        resize_on_border=true;
      };
      animations = {
        "bezier" = "myBezier, 0.05, 0.9, 0.1, 1.05";
      }
    }
  }
  "animation" = [
    "windows, 1, 7, myBezier"
    "windowsOut, 1, 7, default, popin 80%"
    "border, 1, 10, default"
    "borderangle, 1, 8, default"
    "fade, 1, 7, default"
    "workspaces, 1, 6, default"
  ]
}
```

```
[];  
[enabled = true;  
  };  
decoration = {  
  rounding = 10;  
  active_opacity=1.0;  
  "col.shadow"="rgba(1a1a1aee)";  
  drop_shadow=true;  
  inactive_opacity=1.0;  
  shadow_range=4;  
  shadow_render_power=3;  
  blur = {  
    "enabled"="true";  
    "passes"="1";  
    "size"="3";  
    "vibrancy"="0.169600";  
  };  
};  
dwindle = {  
  preserve_split=true;  
  pseudotile=true;  
};  
master = {  
  new_status="master";  
};  
input = {  
  touchpad = {  
    natural_scroll=true;  
  };  
  follow_mouse=1;  
  kb_options="ctrl:nocaps";  
  sensitivity=0;  
};  
env = [  
  "XCURSOR_SIZE,38"  
  "HYPRCURSOR_SIZE,38"  
  "BEMENU_BACKEND,wayland"  
];  
  
bind = [
```

"\$mainMod, Return, exec, \$terminal"
"\$mainMod, Q, killactive"
"\$mainMod SHIFT, E, exit"
"\$mainMod, W, exec, \$browser"
"\$mainMod SHIFT, Space, togglefloating"
"\$mainMod SHIFT, D, exec, \$menu"
"\$mainMod, F, fullscreen, 1"
"\$mainMod, E, togglegroup"

□"\$mainMod, S, changegroupactive"

"\$mainMod SHIFT, minus, movetoworkspacesilent, special"
"\$mainMod, minus, togglespecialworkspace, special"
"\$mainMod, h, movefocus, l"
"\$mainMod, l, movefocus, r"
"\$mainMod, k, movefocus, u"
"\$mainMod, j, movefocus, d"
"\$mainMod, 1, workspace, 1"
"\$mainMod, 2, workspace, 2"
"\$mainMod, 3, workspace, 3"
"\$mainMod, 4, workspace, 4"
"\$mainMod, 5, workspace, 5"
"\$mainMod, 6, workspace, 6"
"\$mainMod, 7, workspace, 7"
"\$mainMod, 8, workspace, 8"
"\$mainMod, 9, workspace, 9"
"\$mainMod, 0, workspace, 10"
"\$mainMod SHIFT, 1, movetoworkspace, 1"
"\$mainMod SHIFT, 1, workspace, 1"
"\$mainMod SHIFT, 2, movetoworkspace, 2"
"\$mainMod SHIFT, 2, workspace, 2"
"\$mainMod SHIFT, 3, movetoworkspace, 3"
"\$mainMod SHIFT, 3, workspace, 3"
"\$mainMod SHIFT, 4, movetoworkspace, 4"
"\$mainMod SHIFT, 4, workspace, 4"
"\$mainMod SHIFT, 5, movetoworkspace, 5"
"\$mainMod SHIFT, 5, workspace, 5"
"\$mainMod SHIFT, 6, movetoworkspace, 6"
"\$mainMod SHIFT, 6, workspace, 6"
"\$mainMod SHIFT, 7, movetoworkspace, 7"
"\$mainMod SHIFT, 7, workspace, 7"
"\$mainMod SHIFT, 8, movetoworkspace, 8"

```

"$mainMod SHIFT, 8, workspace, 8"
"$mainMod SHIFT, 9, movetoworkspace, 9"
"$mainMod SHIFT, 9, workspace, 9"
"$mainMod SHIFT, 0, movetoworkspace, 10"
"$mainMod SHIFT, 0, workspace, 10"
"$mainMod SHIFT, h, movewindoworgroup, l"
"$mainMod SHIFT, j, movewindoworgroup, d"
"$mainMod SHIFT, k, movewindoworgroup, u"
"$mainMod SHIFT, l, movewindoworgroup, r"
"$mainMod, mouse_down, workspace, e+1"
"$mainMod, mouse_up, workspace, e-1"
", XF86AudioPlay, exec, playerctl play-pause"
", XF86AudioNext, exec, playerctl next"
", XF86AudioPrev, exec, playerctl previous"
", XF86Tools, exec, playerctl play-pause"
", XF86Favorites, exec, playerctl next"
", XF86Bluetooth, exec, playerctl previous"
", XF86MonBrightnessUp, exec, brightnessctl set +10%"
", XF86MonBrightnessDown, exec, brightnessctl set --min-value=1 10%-"
", XF86AudioRaiseVolume, exec, wpctl set-volume -l 1.5 @DEFAULT_SINK@ 5%+ && wpctl set-mute
@DEFAULT_SINK@ 0"
", XF86AudioLowerVolume, exec, wpctl set-volume @DEFAULT_SINK@ 5%- && wpctl set-mute
@DEFAULT_SINK@ 0"
", XF86AudioMute, exec, wpctl set-mute @DEFAULT_SINK@ toggle"
", XF86AudioMicMute, exec, wpctl set-mute @DEFAULT_SOURCE@ toggle"
"$mainMod&Alt_L, k, exec, hyprctl keyword monitor ,preferred,auto,1,transform,0"
"$mainMod&Alt_L, k, exec, hyprctl keyword input:touchdevice:transform 0"
"$mainMod&Alt_L, k, exec, hyprctl keyword input:tablet:transform 0"
"$mainMod&Alt_L, h, exec, hyprctl keyword monitor ,preferred,auto,1,transform,1"
"$mainMod&Alt_L, h, exec, hyprctl keyword input:touchdevice:transform 1"
"$mainMod&Alt_L, h, exec, hyprctl keyword input:tablet:transform 1"
"$mainMod&Alt_L, j, exec, hyprctl keyword monitor ,preferred,auto,1,transform,2"
"$mainMod&Alt_L, j, exec, hyprctl keyword input:touchdevice:transform 2"
"$mainMod&Alt_L, j, exec, hyprctl keyword input:tablet:transform 2"
"$mainMod&Alt_L, l, exec, hyprctl keyword monitor ,preferred,auto,1,transform,3"
"$mainMod&Alt_L, l, exec, hyprctl keyword input:touchdevice:transform 3"
"$mainMod&Alt_L, l, exec, hyprctl keyword input:tablet:transform 3"
];
bindm = [
"$mainMod, mouse:272, movewindow"

```

```
"$mainMod, mouse:273, resizingwindow"
];
}; #Settings block
extraConfig = "
    bind=$mainMod,R,submap,resize
    submap=resize
    binde=,l,resizeactive,150 0
    binde=,h,resizeactive,-150 0
    binde=,k,resizeactive,0 -150
    binde=,j,resizeactive,0 150

    bind=,escape,submap,reset
    submap=reset

    $mode-system = (k) lock | (e) exit | (s) poweroff | (r) reboot
    bind = $mainMod, x, submap, $mode-system
    submap = $mode-system
    bind = , k, exec, hyprlock --immediate
    bindl = , k, submap, reset
    bind = , e, exit
    bind = , r, exec, systemctl reboot
    bind = , s, exec, systemctl poweroff
    bind = , escape, submap, reset
    submap = reset
";
};
}
```

hyprlock.nix

```
{ config, pkgs, lib, ... }:  
  
{  
  programs.hyprlock = {  
    enable = true;  
    settings = {  
      general = {  
        disable_loading_bar = true;  
        grace = 300;  
        hide_cursor = true;  
        no_fade_in = false;  
      };  
  
      background = [  
        {  
          path = "screenshot";  
          blur_passes = 3;  
          blur_size = 8;  
        }  
      ];  
  
      input-field = [  
        {  
          size = "200, 50";  
          position = "0, -80";  
          monitor = "";  
          dots_center = true;  
          fade_on_empty = false;  
          font_color = "rgb(202, 211, 245)";  
          inner_color = "rgb(91, 96, 120)";  
          outer_color = "rgb(24, 25, 38)";  
          outline_thickness = 5;  
          placeholder_text = "<span foreground='##cad3f5'>Password...</span>";
```

```
        shadow_passes = 2;
    }
};
};
}
```

kitty.nix

```
{ config, pkgs, lib, ... }:  
  
{  
  programs.kitty = lib.mkForce {  
    enable = true;  
    settings = {  
      confirm_os_window_close = 0;  
      dynamic_background_opacity = true;  
      background_opacity = "0.5";  
      background_blur = 5;  
      font_family = "Noto Sans";  
    };  
  };  
};  
}
```

neovim.nix

```
{ config, pkgs, lib, ... }:  
  
{  
  programs.neovim = {  
    enable = true;  
    extraConfig = "  
      set number relativenumber  
    ";  
    viAlias = true;  
    vimAlias = true;  
    defaultEditor = true;  
  };  
}
```

waybar.nix

```
{ config, pkgs, lib, ... }:

{
  programs.waybar = {
    enable = true;
    settings = {
      mainBar = {
        layer = "top";
        position = "top";
        modules-left = [ "hyprland/workspaces" "hyprland/submap" ];
        modules-center = [ "mpris" ];
        modules-right = [ "backlight" "privacy" "wireplumber" "disk" "load" "memory" "battery" "tray" "clock" ];

        "hyprland/workspaces" = {
          all-outputs = true;
          on-click = "activate";
        };

        "wireplumber" = {
          "format" = "{icon} {volume}%";
          "format-muted" = "[]";
          "on-click" = "set-mute @DEFAULT_SINK@ toggle";
          "max-volume" = 150;
          "scroll-step" = 0.2;
        };
        format-icons = [ "[]" "[]" "[]" ];
      };
      disk = {
        format = "[{specific_free:0.0f}GB";
        unit = "GB";
      };
      "mpris" = {
        "format" = "{player_icon} {title}";
        "format-paused" = "{status_icon} <i>{title}</i>";
        "player-icons" = {
```

```

    "default" = "□";
    "mpv" = "□";
};
"status-icons" = {
    "paused" = "□";
};
max-length = 50;
"ignored-players" = [ "firefox" ];
};
"clock" = {
    "tooltip-format" = "<tt><small>{calendar}</small></tt>";
    "format" = "□ {:%H:%M %m/%d/%Y}";
};
"calendar" = {
    "mode" = "year";
    "mode-mon-col" = 3;
    "weeks-pos" = "right";
    "on-scroll" = 1;
    "format" = {
        "months" = "<span color='#ffead3'><b>{ }</b></span>";
        "days" = "<span color='#ecc6d9'><b>{ }</b></span>";
        "weeks" = "<span color='#99ffdd'><b>W{ }</b></span>";
        "weekdays" = "<span color='#ffcc66'><b>{ }</b></span>";
        "today" = "<span color='#ff6699'><b><u>{ }</u></b></span>";
    };
};
"actions" = {
    "on-click-right" = "mode";
    "on-click-forward" = "tz_up";
    "on-click-backward" = "tz_down";
    "on-scroll-up" = "shift_up";
    "on-scroll-down" = "shift_down";
};
"backlight" = {
    "device" = "intel_backlight";
    "format" = "{icon}";
    "format-icons" = ["□" "□" "□" "□" "□" "□" "□" "□"];
};
"battery" = {
    "states" = {

```

```

        "warning" = 30;
        "critical" = 15;
    };
    "format-icons" = [ "⚡" "🔋" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" "🔌" ];
    "format" = "{icon} {capacity}%";
    # "format-charging" = "⚡ {capacity}%";
    # "format-plugged" = "🔌 {capacity}%";
    "format-alt" = "{time}";
};
};
};
style = "
@import "/home/.home-manager/mocha.css";

* {
    font-family: Fira Code;
    font-size: 17px;
    min-height: 0;
}

#waybar {
    background: transparent;
    color: @text;
    margin: 5px 5px;
}

#workspaces {
    border-radius: 1rem;
    margin: 5px;
    background-color: @surface0;
    margin-left: 1rem;
}

#workspaces button {
    color: @lavender;
    border-radius: 1rem;
    padding: 0.4rem;
}

#workspaces button.active {

```

```
color: @sky;
border-radius: 1rem;
}
```

```
#workspaces button:hover {
  color: @sapphire;
  border-radius: 1rem;
}
```

```
#custom-music,
#backlight,
#privacy,
#network,
#wireplumber,
#disk,
#load,
#memory,
#battery,
#tray,
#clock {
  background-color: @surface0;
  padding: 0.5rem 1rem;
  margin: 5px 0;
}
```

```
#clock {
  color: @blue;
  border-radius: 0px 1rem 1rem 0px;
  margin-right: 1rem;
}
```

```
#battery {
  color: @green;
}
```

```
#battery.charging {
  color: @green;
}
```

```
#battery.warning:not(.charging) {
```

```
    color: @red;
}

#backlight {
    color: @yellow;
    margin-left: 1rem;
    border-radius: 1rem 0px 0px 1rem;
}

#wireplumber {
    color: @maroon;
}

#tray {
    border-radius: 0px 0px 0px 0px
}
";
};
}
```

mocha.css

```
@define-color rosewater #f5e0dc;
@define-color flamingo #f2cdcd;
@define-color pink #f5c2e7;
@define-color mauve #cba6f7;
@define-color red #f38ba8;
@define-color maroon #eba0ac;
@define-color peach #fab387;
@define-color yellow #f9e2af;
@define-color green #a6e3a1;
@define-color teal #94e2d5;
@define-color sky #89dceb;
@define-color sapphire #74c7ec;
@define-color blue #89b4fa;
@define-color lavender #b4befe;
@define-color text #cdd6f4;
@define-color subtext1 #bac2de;
@define-color subtext0 #a6adc8;
@define-color overlay2 #9399b2;
```

```
@define-color overlay1 #7f849c;  
@define-color overlay0 #6c7086;  
@define-color surface2 #585b70;  
@define-color surface1 #45475a;  
@define-color surface0 #313244;  
@define-color base #1e1e2e;  
@define-color mantle #181825;  
@define-color crust #11111b;
```

zsh.nix

```
{ config, lib, pkgs, ... }:

{
  programs.zsh = {
    enable = true;
    enableCompletion = true;
    autosuggestion.enable = true;
    syntaxHighlighting.enable = true;
  };
}
```

tmux.conf

```
"

set -g mouse on
set-window-option -g mode-keys vi


unbind C-B
set -g prefix C-Space
bind C-Space send-prefix


bind R source-file ~/.tmux.conf \; display "Reloaded!"


set-option -g history-limit 50000


bind C-j select-pane -D
bind C-k select-pane -U
bind C-h select-pane -L
bind C-l select-pane -R
bind j select-pane -D
bind k select-pane -U
bind h select-pane -L
bind l select-pane -R


bind r switch-client -T RESIZE


bind -T RESIZE k resize-pane -U \; switch-client -T RESIZE
bind -T RESIZE j resize-pane -D \; switch-client -T RESIZE
bind -T RESIZE h resize-pane -L \; switch-client -T RESIZE
bind -T RESIZE l resize-pane -R \; switch-client -T RESIZE


bind -T RESIZE K resize-pane -U 5 \; switch-client -T RESIZE
bind -T RESIZE J resize-pane -D 5 \; switch-client -T RESIZE
bind -T RESIZE H resize-pane -L 5 \; switch-client -T RESIZE
bind -T RESIZE L resize-pane -R 5 \; switch-client -T RESIZE
```

```
bind s split-window -c "#{pane_current_path}"
```

```
bind v split-window -h -c "#{pane_current_path}"
```

```
bind Space new-window -c "#{pane_current_path}"
```

```
bind C-Space new-window -c "#{pane_current_path}"
```

```
unbind x
```

```
bind c confirm-before -p "kill-pane #W? (y/n)" kill-pane
```

```
bind N previous-window
```

```
bind y switch-client -T copy-mode
```

```
""
```

nixvim.nix

```
let
nixvim = import (builtins.fetchGit {
  url = "https://github.com/nix-community/nixvim";
});
in
{
  imports = [
    nixvim.homeManagerModules.nixvim
  ];
  programs.nixvim = {
    enable = true;
    defaultEditor = true;
    vimdiffAlias = true;
    viAlias = true;
    vimAlias = true;
    opts = {
      number = true;
      relativenumber = true;
      shiftwidth = 2;
    };
    plugins = {
      lualine.enable = true;
      bufferline.enable = true;
      nvim-tree.enable = true;
      telescope = {
        enable = true;
        extensions = {
          file-browser.enable = true;
        };
      };
      keymaps = {
        "<C-p>" = {
          action = "git_files";
          options = {
```

```
    desc = "Telescope Git Files";
  };
};
  "<leader>fg" = "live_grep";
};
  };
  indent-blankline.enable = true;
  cmp.enable = true;
  lsp = {
enable = true;
servers = {
  nixd.enable = true;
};
  };
  };
  colorschemes.gruvbox.enable = true;
};
}
```

shared-home.nix

```
{ config, pkgs, lib, inputs, ... }:

{
  imports = [
    ./bemenu.nix
    ./hypridle.nix
    ./hyprland.nix
    ./hyprlock.nix
    ./kitty.nix
    # ./neovim.nix
    ./nixvim.nix
    ./waybar.nix
    ./zsh.nix
  ];

  home.stateVersion = "24.05"; # Please read the comment before changing.

  home.packages = [
    pkgs.grim
    pkgs.slurp
    pkgs.playerctl
    pkgs.brightnessctl
    pkgs.wl-clipboard
    pkgs.qrencode
    pkgs.libarchive
  ];

  home.file = { };

  home.sessionVariables = { };

  xdg.userDirs = {
    enable = true;
```

```
createDirectories = true;
};

programs.home-manager.enable = true;
programs.xplr.enable = true;
programs.imv.enable = true;
programs.bat.enable = true;
programs.zathura = {
  enable = true;
  extraConfig = "
    set selection-clipboard clipboard
    set sandbox strict
  ";
};

services.playerctld.enable = true;

}
```