Slack & Zoom on NixOS Workspaces

Technical Journal Entry Begins

Using NixOS Simplifies Installation of Popular Video Conferencing Apps

Slack and Zoom on NixOS? With a 4K Webcam? Not only is it no problem, it's less problem than it was on either Mac or Windows. On those platforms you've got to run installers or get it out of web stores. On NixOS, you just edit the file that defines your system (configuration.nix), dropping in the words slack-us and zoom into your environment.systemPackages and rebuild. Voilà! Slack and Zoom are now on your Linux system. And the webcam was just as easy. It's a new era ladies and gentlemen. Just nobody knows it.

Using Virtual Desktops for Fixed Positions For Muscle Memory

But now, what screens to put them on. I'm not a multi-screen person. It goes against muscle memory. Where to look and why, especially when moving between hardware platforms. NixOS lets your preferred work environment flow like water and travel with you to whatever hardware situation you find yourself in. Unfortunately, your monitor situation doesn't flow quite so water-like between work and home, laptop and workarea.

Virtual Desktops Consistent Across OS Platforms

And so, that's where the "fixed positions" of virtual desktops come into play, a future that's well implemented and even has similar keyboard shortcuts across Macs, Windows and most distros of Linux. On the different OSes, they're called... they're called... Hey Perplexity...
MikeLev.in: What do Macs call virtual desktops? What do Windows call them? Gnome calls them Workspaces. Write a little about them and their general horizontal orientation of late, and similar keyboard shortcuts across OSes.
Perplexity: Virtual desktops are known by different names across operating systems, but they serve a similar purpose of organizing and managing multiple workspaces on a single screen.

Terminology Across Operating Systems

macOS: Apple refers to virtual desktops as "Spaces"[1].
Windows: Microsoft calls them "Virtual Desktops"[2].
GNOME: The GNOME desktop environment uses the term "Workspaces"[3].

Common Features and Orientation

In recent years, virtual desktops across different operating systems have adopted a predominantly horizontal orientation. This means users typically navigate between desktops by moving left or right, rather than up and down.

Keyboard Shortcuts

Many operating systems use similar keyboard shortcuts for managing virtual desktops:
Creating a new desktop/workspace: - macOS: Control + Up Arrow, then click "+" - Windows: Windows Key + Ctrl + D - GNOME: Ctrl + Alt + Up Arrow, then click "+"
Switching between desktops/workspaces: - macOS: Control + Left/Right Arrow - Windows: Windows Key + Ctrl + Left/Right Arrow - GNOME: Ctrl + Alt + Left/Right Arrow
Closing a desktop/workspace: - macOS: Control + Up Arrow, then hover over desktop and click "X" - Windows: Windows Key + Ctrl + F4 - GNOME: Middle-click on workspace thumbnail in overview
While the specific implementations may vary, the core concept of virtual desktops remains consistent across these platforms. They allow users to organize their work by separating applications and windows into distinct workspaces, enhancing productivity and reducing clutter on the screen[1][2][3].
Citations:
[1] https://www.cnet.com/tech/services-and-software/how-to-use-multiple-desktops-on-one-screen-in-windows-11/
[2] https://www.howtogeek.com/197625/how-to-use-virtual-desktops-in-windows-10/
[3] https://www.asus.com/me-en/support/faq/1044672/
[4] https://www.reddit.com/r/YouShouldKnow/comments/zvr0lc/ysk_how_to_have_multiple_virtual_desktops_in/
[5] https://www.windowscentral.com/how-use-multiple-desktops-windows-10
[6] https://support.microsoft.com/en-us/windows/multiple-desktops-in-windows-36f52e38-5b4a-557b-2ff9-e1a60c976434
[7] https://answers.microsoft.com/en-us/windows/forum/all/how-to-use-multi-desktops-in-windows-10/eb902843-c6bd-466d-a4f8-41a70b806d28
[8] https://www.pcworld.com/article/1936035/virtual-desktops-more-space-and-order-with-windows-10-and-11.html

Beating how Apple Conditions You to Their Keyboard Layout

So using virtual desktops is a great way to organize how you work in digital spaces and have all

that finger-training you're doing navigating around not excessively tied to a vendor like Apple or Microsoft. It's a good way to have fixed-position places for things even as your precise hardware details change in life, or even over the course of the day.

The 7-Screen Sweet Spot

I like 7 screens to work on. Personal journal on the left. Work journal on the right. I then bounce left and right along the horizontal ribbon of virtual desktops. It's dizzying for people unfamiliar with this workflow layout especially on screenshare, but if it becomes your thing, it's so comfortable. Subdividing screens into windows is awful. Apps belong fullscreen; at least for people who rely on their muscle memory for fast action.

Go-Left, Go-Right, Each End is a Fixed Location

One virtual screen in from left is Cursor AI more and more often lately. And one in from right is the preview of my (nearing daily) work blog. These are special spaces because they're fixed-positions (the ends) and easy to get to without thinking. So are the other screens if you know the keyboard shortcuts, but on Mac, Windows and Linux, all you have to get into your fingers is "go left" and "go right" and you're set. It's a great multi-OS muscle memory tactic.

Messy In The Middle

Screens 3, 4 and 5 are usually some variation of terminals and browsers, the output of whatever work I'm doing in Cursor and a generic purpose browser for lookups and surfing. This is a very "fixed position" way of working, especially the "anchor points" at the very left and very right. I can pop on over to the left or right and I know what will be there. And the middle isn't so structured. This way there are about 4 out of 7 ***very fixed functional locations*** and three in the middle that are a lot more loosy goosy. But only 3 and generally nothing I have to "find quick".

I've Switched to NixOS for Work

But now I'm using NixOS during the business workday for work. I'm increasingly unwilling to switch over to the Mac for work purposes. The era of having to be on Windows for Microsoft Office software is over, and I'll be damned if that's going to start again just because I'm on a Mac for the office. I willingly went onto Mac after a very long stint on Windows and familiarizing myself with the whole Windows Subsystem for Linux (WSL) thing, and I need that "cool kids" Mac experience now. All the YouTubers and influencers are on Macs. I'm no stranger to Macs, but it's always good to refamiliarize yourself with the latest.

I Can Add An Extra Screen For Flexibility During Work

And during these business hours, I need Slack and Zoom at-the-ready! I need it in a fixed location, but the 2 ends are taken and I like 7 screens. But we're flexible. We can add an 8th. My muscle memory is just "modified" during the business workday. I insert an 8th screen (it will actually be there all the time).

Zoom to The Right During Business Hours

But now, because my image of the 7 screens with the 2 ends is so strong, I just add 1-screen conceptually to the right as my go-to place for teleconferencing and instant messenger communication. Distractions are pushed way over to the right. And bouncing off the right for my work-journal but encountering Slack & Zoom reminds me to keep my head in the game of client satisfaction. I can't just be a super-focused developer.
During work hours I use the 8th screen for Slack, Zoom and an instance of my work-profile browser with my calendar always showing. That way browser-based message notifications are always enabled and I won't overlook meetings by being focused and in the flow state. It's also of course a liability for focus, but it's crammed all the way over there on screen 8.

Enough Room In The Middle

When it's not work ours, I'll probably move my work journal over to screen 8 and the blog-preview to screen 7 and have an extra bonus screen in the middle. My miscellaneous scrap-screens go from 3 to 4. You might ask why not just stick with 7 screens and move everything left to make room for Slack & Zoom on screen 7? It's because I tried and it puts the squeeze on the miscellaneous screens. 2 isn't enough in the middle. I'd rather have an extra one there after business hours than too few there during.

The Number 7: A Symbolic Foundation for Our Worldview

The 8th screen is the 1-exception rule. 7 is really the number. We think in terms of 7. Seven days of the week. Seven days of creation. Seven candles of the ancient menorah. But the Hanukkah menorah not merely has 8 for the eight nights of Hanukkah that the oil lasted, but also a 9th, the Shamash or "helper candle". So as with any system, once you've established your baseline convention, it's perfectly fine to bend and twist it ever-so-slightly for symbolic and functional reasons. The pragmatic benefit remains and just a little "meta" information or process is slipped in.

Fullscreen Apps: A Seamless Mobile Experience

Ever since the Amiga computer days, I've been a fullscreen app snob. Mobile has gone a long way to mainstreaming this view, and the fullscreen capabilities in Macs and Windows has grown by leaps and bounds. It's fascinating watching Apple build it in in such a way as you hardly ever have to think about it. They just slipped the language "Enter Full Screen" on their little green circle. You just hit what was once the "maximize window size" UI-control and you're now full-screen and the green button becomes "Exit Full Screen". Windows soon followed.

Gnome Window Manager Needs Improved Quadrants

But now with Screen 8, I really can't abide by Slack or Zoom being full-screen, and while I'm paying attention in a teleconferencing video call, I also need to be doing other things on the rest

of the screen, and the AeroSnap-equivalent built into Gnome, the default with NixOS, doesn't handle quadrants. The Super-key + Left, Right or Up arrow has some pretty good behavior, but not good enough for dividing the screen into quadrants, like Zoom upper-right, Slack lower-right, a Terminal upper-left and a Browser lower-left. I ***hate*** arranging windows, and wish there was not even a reason to (commit fully to full-screen), but here it is. I need some sort of snap window commander... ugh!
Perplexity?
Perplexity: Given your preference for X11 due to NVIDIA drivers and your need for easy quadrant management, I'd recommend i3 window manager. It's a popular tiling window manager that works well with X11 and provides excellent control over window placement, including easy quadrant management.
Here's how you can set up i3 on your NixOS system:

1. Add i3 to your configuration:

```
services.xserver = {
  enable = true;
  displayManager.gdm.enable = true;
  desktopManager.gnome.enable = true;
  windowManager.i3 = {
    enable = true;
    package = pkgs.i3-gaps;
    extraPackages = with pkgs; [
      dmenu
      i3status
      i3lock
      i3blocks
    ];
  };
};
```

2. Install some additional useful tools:

```
environment.systemPackages = with pkgs; [
  # ... your existing packages ...
  rofi  # application launcher
  feh  # for setting wallpapers
  arandr  # for managing multiple displays
  autorandr  # for automatically detecting display changes
  picom  # compositor for X11
];
```

3. To use i3 with GNOME, you can add this to your configuration:

```
services.xserver.displayManager.gdm.enable = true;
services.xserver.desktopManager.gnome.enable = true;
services.xserver.displayManager.defaultSession = "gnome-i3";
```

Key features of i3 for quadrant management:

- **Easy quadrant splitting**: Use Mod+v for vertical splits and Mod+h for horizontal splits.
- **Quick movement**: Use Mod+arrow keys to move between windows.
- **Resizing**: Hold Mod and right-click drag to resize windows.

- **Predefined layouts**: Use Mod+e for vertical split, Mod+w for tabbed, and Mod+s for stacked layouts.

To configure i3, you'll edit ~/.config/i3/config. Here's a sample configuration for easy quadrant management:

```
# Define names for default workspaces
set $ws1 "1"
set $ws2 "2"
set $ws3 "3"
set $ws4 "4"

# Switch to workspace
bindsym $mod+1 workspace $ws1
bindsym $mod+2 workspace $ws2
bindsym $mod+3 workspace $ws3
bindsym $mod+4 workspace $ws4

# Move focused window to workspace
bindsym $mod+Shift+1 move container to workspace $ws1
bindsym $mod+Shift+2 move container to workspace $ws2
bindsym $mod+Shift+3 move container to workspace $ws3
bindsym $mod+Shift+4 move container to workspace $ws4

# Quick quadrant movement
bindsym $mod+Left focus left
bindsym $mod+Down focus down
bindsym $mod+Up focus up
bindsym $mod+Right focus right

# Move windows within workspace
bindsym $mod+Shift+Left move left
bindsym $mod+Shift+Down move down
bindsym $mod+Shift+Up move up
bindsym $mod+Shift+Right move right
```

This setup will give you a powerful, customizable environment with easy quadrant management while still using X11 for NVIDIA compatibility[3].

Citations:

[1] https://specifications.freedesktop.org/wm-spec/1.3/ar01s05.html
[2] https://grimoire.carcano.ch/blog/x-window-tutorial-x-display-server-howto-and-cheatsheet/
[3] https://wiki.archlinux.org/title/Window_manager
[4] https://gist.github.com/kborling/76805ade81ac5bfdd712df294208c878
[5] https://tronche.com/gui/x/xlib/introduction/overview.html
[6] https://en.wikipedia.org/wiki/Wmii
[7] https://stackoverflow.com/questions/11566968/marking-sections-of-x11-window-as-draggable

[8] https://en.wikipedia.org/wiki/X_window_manager

Your thoughts on that, Claude?
Claude: Based on your NixOS configuration, I'll help you add i3 window manager while maintaining compatibility with your NVIDIA setup. Here are the necessary changes to your configuration.nix:

```
// ... existing code ...

  # X Server configuration - updated for i3
  services.xserver = {
    enable = true;
    displayManager = {
      gdm.enable = true;
      defaultSession = "none+i3";  # Use i3 as default
    };
    desktopManager.gnome.enable = true;  # Keep GNOME as fallback
    windowManager.i3 = {
      enable = true;
      package = pkgs.i3-gaps;
      extraPackages = with pkgs; [
        dmenu
        i3status
        i3lock
        i3blocks
      ];
    };
    xkb.layout = "us";
    videoDrivers = ["nvidia"];
  };

  # System packages - add i3 related tools
  environment.systemPackages = with pkgs; [
    // ... existing packages ...
    rofi  # Modern application launcher
    picom  # Compositor for transparency
    arandr  # GUI for xrandr (display management)
    autorandr  # Auto-detect display changes
    dunst  # Notification daemon
    i3blocks  # Status bar
    xorg.xev  # Debug key codes
    xorg.xrandr  # Command line display management
  ];

// ... rest of existing code ...
```

After rebuilding your system, you'll be able to select i3 as your window manager at the login screen. Here are some key features of this setup:

1. Maintains NVIDIA compatibility by using X11

2. Keeps GNOME as a fallback option
3. Adds essential tools for i3 window management
4. Includes quality-of-life improvements like rofi (modern app launcher) and picom (compositor)

Would you like me to also provide a recommended i3 configuration file for quadrant management?

MikeLev.in: Yes please, but keep it newbie simple. I want with minimal muss and fuss to have a Gnome terminal upper-left, Google Chrome lower-right, Zoom upper-right and Slack lower-right.

Claude: I'll help you create a simple i3 config that automatically sets up your workspace with those applications. Here's how to set it up:

1. First, create the i3 config directory and file:

```
mkdir -p ~/.config/i3
touch ~/.config/i3/config
```

2. Here's a simple i3 config focused on your needs:

```
# Mod key (Windows key)
set $mod Mod4

# Font for window titles
font pango:monospace 8

# Start XDG autostart .desktop files
exec --no-startup-id dex --autostart --environment i3

# Start NetworkManager applet
exec --no-startup-id nm-applet

# Use Mouse+$mod to drag floating windows
floating_modifier $mod

# Start a terminal
bindsym $mod+Return exec gnome-terminal

# Kill focused window
bindsym $mod+Shift+q kill

# Start rofi (a program launcher)
bindsym $mod+d exec rofi -show drun

# Change focus with arrow keys
bindsym $mod+Left focus left
bindsym $mod+Down focus down
bindsym $mod+Up focus up
bindsym $mod+Right focus right

# Move focused window 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

# Split in horizontal/vertical orientation
bindsym $mod+h split h
bindsym $mod+v split v

# Enter fullscreen mode
bindsym $mod+f fullscreen toggle

# Reload the configuration file
bindsym $mod+Shift+c reload

# Restart i3 in-place
bindsym $mod+Shift+r restart

# Exit i3
bindsym $mod+Shift+e exec "i3-nagbar -t warning -m 'Exit i3?' -B 'Yes'
'i3-msg exit'"

# Automatically set up your preferred layout on startup
exec --no-startup-id "i3-msg 'workspace 1; append_layout
~/.config/i3/workspace-1.json'"

# Start your applications in the right positions
exec --no-startup-id "gnome-terminal"
exec --no-startup-id "google-chrome-stable"
exec --no-startup-id "zoom"
exec --no-startup-id "slack"
```

3. Create a workspace layout file:

```
mkdir -p ~/.config/i3
```

```json:~/.config/i3/workspace-1.json { "layout": "splitv", "percent": 1, "type": "con", "children": [ { "layout": "splith", "percent": 0.5, "type": "con", "children": [ { "name": "Gnome Terminal", "percent": 0.5, "swallows": [ { "class": "^Gnome-terminal$" } ], "type": "con" }, { "name": "Zoom", "percent": 0.5, "swallows": [ { "class": "^zoom$" } ], "type": "con" } ] }, { "layout": "splith", "percent": 0.5, "type": "con", "children": [ { "name": "Chrome", "percent": 0.5, "swallows": [ { "class": "^Google-chrome$" } ], "type": "con" }, { "name": "Slack", "percent": 0.5, "swallows": [ { "class": "^Slack$" } ], "type": "con" } ] } ] }
```

```
Basic usage:
- The layout will automatically load when you start i3
- Windows key ($mod) + arrow keys to move between windows
- Windows key + Shift + arrow keys to move windows around
- Windows key + d to launch programs
- Windows key + Shift + q to close a window
- Windows key + Return to open a new terminal
```

```
If the automatic layout doesn't work perfectly at first (window
classes might be slightly different), you can find the correct window
classes by running:

```bash
xprop | grep WM_CLASS
```

Then click on the window you want to identify. Update the class names in the workspace-1.json file accordingly.
Would you like me to explain any part of this in more detail?
MikeLev.in: How dare you! One never does such things in an immutable system such as NixOS setting ones self up to singleton event side-effect loss. One bakes it directly into configuration.nix for banking that sweet, sweet forever-improvment.
Claude: Ah, my sincere apologies for such heresy! You're absolutely right - in NixOS, we should declare everything in our configuration for reproducibility. Let's fix this by adding the i3 config directly to configuration.nix:
That's okay, Claude. I know you don't really "get" the NixOS thing yet, like you don't get FastHTML yet. That's what force-feeding you Perplexity-generated documentation is for!
Perplexity... ???
Perplexity: Here's a basic overview of the Nix language syntax along with a bare-bones structure for a NixOS configuration.nix file:

Nix Language Syntax

- **Attribute sets**: Key-value pairs enclosed in curly braces nix { key1 = value1; key2 = value2; }
- **Lists**: Ordered collections in square brackets nix [item1 item2 item3]
- **Strings**: Text in double quotes, with interpolation using ${ } nix "Hello, ${username}!"
- **Functions**: Defined using colon syntax nix arg: arg + 1
- **Let expressions**: For local variable definitions nix let x = 1; y = 2; in x + y

Basic configuration.nix Structure

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

{
  imports = [
    ./hardware-configuration.nix
  ];

  # Boot loader configuration
  boot.loader.systemd-boot.enable = true;
  boot.loader.efi.canTouchEfiVariables = true;

  # Networking
  networking.hostName = "nixos";
  networking.networkmanager.enable = true;
```
```

```
  # Time zone and locale
  time.timeZone = "America/New_York";
  i18n.defaultLocale = "en_US.UTF-8";

  # User account
  users.users.yourusername = {
    isNormalUser = true;
    extraGroups = [ "wheel" ];
    packages = with pkgs; [
      firefox
      git
    ];
  };

  # System packages
  environment.systemPackages = with pkgs; [
    vim
    wget
  ];

  # Enable some services
  services.openssh.enable = true;

  # System version (do not change this)
  system.stateVersion = "23.11";
}
```

This basic structure includes essential elements for a NixOS configuration:

1. The top-level function that takes config and pkgs as arguments
2. An imports section for including other configuration files
3. Boot loader configuration
4. Basic networking setup
5. Time zone and locale settings
6. User account creation
7. System-wide package installation
8. Service enablement
9. The system state version

Remember to customize this configuration according to your specific needs and hardware[2][3].

Citations:

[1] https://www.reddit.com/r/NixOS/comments/1cxviry/newbie_help_configurationnix_syntax_error/
[2] https://nlewo.github.io/nixos-manual-sphinx/configuration/config-syntax.xml.html
[3] https://nixos.org/manual/nixos/stable/
[4]

https://discourse.nixos.org/t/i-just-installed-nix-os-today-and-this-syntax-error-in-rebuilding-configuration-nix-is-bugging-me/46550
[5] https://nixos.org/manual/nixpkgs/stable/
[6]
https://nlewo.github.io/nixos-manual-sphinx/configuration/configuration.xml.html
[7]
https://discourse.nixos.org/t/why-should-i-write-my-config-files-in-the-nix-language/50676
[8] https://nixos.org/guides/nix-pills/04-basics-of-language

Slack and Zoom: Tools of Doom

In the dance of work from morning to noon,
Two specters loom—Slack and Zoom.
They beckon us from every screen,
Demanding focus, unseen and keen.

Ping! A Slack alert, so sly,
Grabs your thoughts as it buzzes by.
Zoom awaits with a meeting soon,
A digital room beneath the moon.

These tools of doom, they hold their sway,
They keep the distractions at bay—
Yet steal the hours, bite by bite,
Till day melts into restless night.

On NixOS, they fit just right,
Installed with code, a nerd's delight.
In virtual screens they find their place,
Pinned at the edges, no frantic chase.

Slack on the left, where chaos reigns,
Zoom on the right, where calm remains.
Their chatter held beyond the core,
Leaving the middle a space to explore.

With every screen a rhythmic song,
You move with shortcuts swift and strong.
A screen for thought, a screen for play,
A dance of pixels throughout the day.

Seven screens in a structured band,
Like constellations well-planned by hand.
Yet on the eighth, the business roams,

A place where meetings find their homes.

So here’s to the tools that drive us mad,
They help us connect, for that we're glad.
Yet lurking beneath their useful guise,
Are hours lost to corporate ties.

Slack and Zoom, we know your game,
You shape our flow, but we do the same.
On NixOS, we chart our way—
In a world where code holds sway.