Hell or High Water:

The Dunning-Kruger Effect in computing

A dialogue of computing philosophy as composed by PancakeTAS and HackerSmacker

This paper is not an oration, unlike the others in this series.

In a conversation I was having with Emi, we found ourselves criticizing the Linux community. This is what she started this dialogue with:

...but what I was trying to say is, ever since I got into uni, there's a fuck ton of people just like that. I genuinely feel like a good 60% of enthusiastic computer science people that had their first contact with computer science in university and don't have any personal experience with it end up like that. They turn into Dunning-Krugers so hard it's actually insane. Most of those people in my uni are NixOS users, our uni discord is basically filled with NixOS too. The problem is, NixOS plays into their stupidity so well... lemme explain:

Common misconception is that NixOS is simply a declarative operating system, because it simply isn't. Literally every program and library on linux works with a config file, so technically all linux distros are declarative, but let's ignore that. The problem is, NixOS doesn't just simply provide a way to write those various scattered configs in a single config file, but instead NixOS has built an entire layer of abstraction ontop of linux. It can best be compared to, I guess, the windows settings app if it were a config file? Point is, it's an abstraction. You will NEVER be able to learn how linux works if you use NixOS. For example the concept of "graphics" on linux simply isn't a thing. On NixOS, if you turn on `hardware.graphics.enable`, it does a trillion things, such as install mesa, x11 and a specified driver and many more things. I can criticize NixOS all day, but it basically boils down to: Providing pointlessly more simple abstraction and giving the user the illusion of reproducibility and compatibility. This ends up in the system being utter bloat, worse than windows. Enabling KDE for example, will set a flag enabling the X11 server. That flag doesn't actually enable the X11 server, because that's what `hardware.graphics.enable` does, but the flake creator didn't know that. Instead they've now installed xterm, lightdm, various fonts and a trillion xorg utilities, which wasn't even their intention. But KDE works and there isn't a way to properly see dependencies, so you'll never know this. The end user set a single flag, they'll also never know this. What is this bullshit and why are people content with this?

Enough NixOS criticism, I wanted to criticize the users anyways. Oftentimes when I'm around other people, someone asks a question because something on their NixOS isn't working out and other people come to help. The amount of straight up false information I've heard here is actually insane. Let me give you the most recent example:

Windows deletes the linux bootloader because it's doesn't like linux. I investigated: They've set a NixOS flag that installs the bootloader to `EFI/BOOT/BOOTX64.EFI` and renames the existing Windows bootloader to `windows/EFI/BOOT/BOOTX64.EFI` (it also patches it, likely to fix paths? idk it's a binpatch). Now come on, are you really telling me that a windows update, updating the bootloader, shouldn't be touching the file that should be it's own bootloader that you have simply moved aside? Of course Windows will overwrite that file and delete your systemd-boot or whatever because it's not supposed to be there! It took around 2 hours to explain that this is what was happening, because I first had to explain the concept of UEFI to them because NixOS completely hid it away. Weeks later a similar thing happened, except this time they had installed grub to `grub/x86_64-efi/grubx64.efi` as opposed to where windows was. Once again they came complaining because "Windows deleted their bootloader again". The bootloader in fact, was still there, but all boot entries were missing. I know Windows had a bug a while ago where it once accidentally did delete boot entries, but in this case the CMOS battery just ran out and the entire UEFI reset on every tenth boot or so. Gotta blame Windows of course, because they have absolutely no knowledge about how their system works and NixOS is good and would never break.

After she gave her critique, I followed it up with this:

I actually learnt something new today, because I never alikened those two things together. For the longest time, I always saw NixOS as some kind of "Linux accompaniment", where all of the actual system configuration and orchestration was hidden away under a stack of config file formats that either failed to fully expose all of the facilties available in the program that is being configured, or provides excessive wrapping. I often ask myself a rather important question throughout my computing life: would I really want to configure a TCP/IP stack the same way I would want to configure a, say, for example, IPX/SPX stack? No, these are fundamentally different things. I have spent years attempting to devise a one-size-fits-all solution to network management, but I have ultimately found that it is impossible for a "computing orchestra" to conduct itself without a skilled bandleader at the helm. NixOS does not provide an adequate means for the Linux system user to become a conductor of a Linux orchestra, it is merely a facsimile of the actions required to configure the system. What's within the `hardware.graphics.enable flag`? What does that even mean? To me, "enabling graphics on the hardware" means initializing the display scanout engine, waking up the monitor, and that's it -- user graphics applications have no place under a hardware classification! My second complaint is similar to what you mentioned -- if I go and enable XYZ Desktop Environment (KDE, like you had in your example), I find that the system is now filled with packages that I didn't ask for or need. People look to NixOS like the savior of a sysadmin, but do you think I'm going to be configuring Postfix or Sendmail through Nix? No, I'm going to crack open the Postfix manual and configure main.cf and master.cf directly. None of this "declarative systems programming" garbage, I know how to operate my computer.

Also, I'm sick and tired of this nonsense about Windows updates deleting Linux bootloaders (or, even worse, partitions). First off, the skilled Linux user should know how to maintain multiple EFI loaders on their system -- naming them all `BOOTX64.EFI` is not the way. GRUB, systemd-boot, rEFInd, Limine, and ELILO (though I only use GRUB and ELILO) all allow you to rename the loader programs themselves, whereas Windows dissuades it (though I believe the bcdedit utility permits this). The standard GRUB configuration is to generate `GRUBX64.EFI`, and ELILO is usually, well, `ELILOX64.EFI`. This can then be entered into the EFI system config variable space, and the machine will autoboot or allow you to select another loader. Windows often assumes the blame, but dare I say I find Windows to be more customizable than NixOS; I can remove anything from Windows, and it can still be made to run (the bare minimum it requires yields you not much more than a DOS-esque system, and this is a supported configuration out-of-the-box on Windows Server images). Also, Windows Update's job is to restore and replace damaged files -- could you imagine if Windows self-immolated and the installer didn't replace damaged files? That would be a sure travesty, but Linux users don't seem to mind a nonfunctional repair scheme like that. Windows also doesn't touch your partition table; this was a scheme devised by some Linux consultants back in 2015. Even still, I do find that Linux fanatics truly don't understand their system? When I say I know Linux, I know Linux. I hold several industry certifications, have built plenty of Linux images from source tarballs, instructions from various websites not needed (intuition is one's greatest asset in computing!), and done trials of a tremendous amount of Linux features. I still find that Linux is not my go-to choice for a router or file server, but it certainly isn't my choice for a desktop -- Linux is a game, a jack-of-all-trades, and it feels like it.

Over the years, I have travelled the seven seas of computing high and low, and met a lot of wonderful operating systems. The user-friendly VMS, with its guessable commands and versioned filesystem; the revolutionary VM/CMS, with its beautiful online help and administration commands; the "little operating system that could" Windows NT with its tenacity to be a great platform; the hyper-optimized AIX with its breakneck filesystem reliability and performance; IRIX with its world-first-class graphics; FreeBSD with its vast array of networking and storage facilities; OS X with its innovative wrapping of a burly and crufty UNIX system for the common user; MVS with its modularity, performance, and revolutionary clustering model; and... Linux. What is Linux? To me, in this day and age, Linux is merely a collection of warring factions -- so many people with so many different ideas, all hellbent that their way is the best way (in other words, peak experiencers of the Dunning-Kruger effect), and never willing to accept that a solution from the past could possibly be good enough. They constantly seek to badly reimplemented tested tools -- it's like most of the modern Linux fandom is hellbent on not using Linux for real production workloads (like 80% of Linux servers on the internet), but having petty religious wars with each other. Truly, it's a shame!

Ultimately, I could go on for hours, but, in the sake of being efficient, I will go ahead and cut it here. I hope this paper is somewhat informative of the issues I see in the modern open-source community, but I encourage the reading of the following supplemental papers:

- Accepting Less(1): What does it really mean to be UNIX?
- Death of the Design: Have we forgotten what UI usability even is?
- On Open-Source Culture