Innovative software comes with a price tag

Disclamer: I'm not promoting or complaining about Linux, macOS or FreeBSD in any way. These are my own opions based on user and developer experience. You are free in whatever you like to get your job done. I don't care.

Linux is not an operating system (OS) as most people think its only a monolithic kernel with many advanced specific technologies. This means the userspace applications and tools come from other places like the GNU project. That is why Linux its is also written as GNU/Linux.

There are exceptions like Alpine Linux try not to use GNU applications and GNU C library (glibc) in their base OS. It feels more like a BSD operating system than a Linux distribution. Alpine uses its own package-management system, apk-tools, which originally was a collection of shell scripts but was later rewritten in C.

I started approximate 15 years ago with a Linux distribution called Knoppix Linux on a 'Live' DVD. A Live distribution can be used without installing it. I bought the DVD on a computer fair for just a few euros. It is a desktop Linux distribution and was created by Klaus Knopper. And to my suprise it is still developed today. It first saw daylight 21 years ago. I experienced some issues with installation on a machine back in the days, but the live DVD was fun to work with. It got me interessted as an alternative to Microsoft Windows.

Time passed after having trouble getting Knoppix running from my harddrive properly. And I bumped upon Ubuntu Linux 5.04 a long time ago. I remember running Ubuntu on my desktop computer until 8.10. Things where very hairy at that time, closed source components where a pain to install (and will always be). It took me a few days during christmas to get the Nvidia graphics card driver compiled for the Linux kernel Ubuntu was using. I had no clue what this fuzz was all about. But you need to install the kernel C headers and must make sure you are downloading the correct Nvidia installer. There must be one part a closed source binary blob and one part which compiles against your running kernel. The Nvidia drivers have been partially reverse engineerd but didn't work that smooth to run a desktop properly.

The Linux kernel is 'designed' in a way that every release can break things very badly. So upgrading your kernel means recompiling your closed source Nvidia driver every damn time. This was very annoying but I got used to it.

I did extensive distro hopping and have had experience with Slackware which is a do it yourself (DIY) distribution and you need to be hardcore to compile and debug stuff on your own. Especially when you want bleeding edge software. This was long before the stackoverflow generation, yes really i'm this old. We've got only mailinglist and the man pages back then. I had no desktop, just a window manager named Fluxbox. And also Openbox which is a clean rewrite of Fluxbox in C++ instead of C. After being annoyed to use a mouse to control my windows I switched once again from 'desktop' software to a tiling window manager which is awesome IMHO so you have full control of your machine from your keyboard only. This was matrix-style computing, and even had my terminal sometimes configured with black or transparent background with green colored text. I felt desktops where bloated and user unfriendly. RSI related issues are mostly due to incorrect mouse use, so I didn't suffered these problems. I felt when not using the mouse I AM the computer. Everything I could think of to control, just needed to use a hotkey and not touching the mouse for extended periods of time. You need real brainpower for this, because all those features behind hotkeys can get complicated. Just like flying an airplane with all those knops and gauges.

Arch Linux was a fairly new distribution which does only rolling releases and never gives out a stable release. Except the onces which are freezed to install media like CD/DVD/USB stick. It ships all software pre-compiled like kernels, libraries and applications. But it is only for power users. It can be run on desktop, server or embedded appliances. The documentation is extensive and available in the form of a wiki which is actively maintained by the community.

In my opinion the Linux desktop is still not widely adopted and failing because the Linux ecosystem is constant in flux and scattered across different communities with very diverged ideas how things should be developed and be used. Desktop systems are complex, and for getting complexity to work API and ABI stability is a must.

Linux desktop is a story on itself. But I try to explain it in a nutshell.

In the open-source world everything is free but usable under the terms of a open-ish license. With so many people on the globe connected to the internet it is natural to have different ideas. They form communities around their own idologies and get followers. The most dominating desktops today are still KDE and GNOME where GNOME suffered a major (community) split between version 2 and 3. This split was caused by version 2 a conventional desktop interface and version 3 a complete rethink of how a desktop could be used. It has a MacOS-like layout, so the idea is basicly stolen in my opinion.

The Linux distribution 'Elementary OS ' is worth mentioning as it looks and feels mimic macOS. And tries to stabilize APIs to build software against. But I have read somewhere the software build for Elementary is hard to run on other distributions.

Parts of this article are based on writing from Wikipedia. So this post is licensed under the CC BY-SA 4.0 which is the same as text on Wikipedia. It is different compared to the rest of my blog which is license under CC0 (public domain). And I heavily rely on linking to Wikipedia for the curious reader.

mindmapping and braindump

- worked on rapidio interconnect subsystem with userspace components -> link to mention in kernel
- 2 years freebsd experience
    - mostly for NAS purposes using ZFS
    - upgrades are fairly smooth, things break but are documented and mentioned or are easy to fix

- divirged landscape of essential software components
- tightly integrated system detects problems early -> api breakages with scattered modules (different communities) are hard to handle
- container debate, just a bunch of complex namespaces and apis
- jails
- badly documented features (kdoc not userspace) -> freebsd has one base userspace and kernel are tightly coupled
- innovation should be managed well -> what does that mean
- the pinguin is getting fat -> lines of code -> complex matrix of kernel build variants
- upgrades to new versions are a pain in embedded systems as every release things get shoveled away or moved elsewhere and this is very badly documented
- linux container namespaces are getting out of hand *

Namespaces

CPU namespace

- https://lwn.net/Articles/872507/
- https://lwn.net/Articles/812504

- A filesystem for namespaces

- ima: Namespace IMA with audit support in IMA-ns

- Controlling the CPU scheduler with BPF

- Linux container schedulers: RunC, systemd-nspawn, AWS bottlerocket, LXC/LXD, Openshift, Kubernetes, containerd, and of course Docker.

- https://containerjournal.com/features/people-want-boycott-docker/

- https://www.slideshare.net/PaoloTonin2/boycott-docker

```
Recently we launched Bottlerocket, a Linux-based container operating s
ystem written in Rust.” — Matt Asay, Amazon Web Services
```