

Migrating from macOS to Linux

Adrian Kosmaczewski

2020-11-07

This is the story of how, after being a loyal macOS user for 15 years, I decided to start using Linux full-time.

This article provides a detailed report of all the small decisions I took from late 2016 until late 2018, the moment in which I got my TUXEDO Computers laptop with Ubuntu 18.04 pre-installed.

- The Situation
 - A Note about Ideology
- 1. Hardware
 - Peripherals
 - Final Choice
- 2. Software
 - Installing Software
 - Note-Taking Apps
 - Web Browser
 - Programming Languages
 - Music
 - Text Editors and IDEs
 - Video
 - Image Editors
 - Password Manager
 - Communications
 - Other Apps
- 3. Data
 - Documents
 - E-Books
- Did Not Happen
- Conclusion

The Situation

For context, I will describe my entanglement with the Apple platform around 2016. Entanglement is the word, yes. The level of intrusion that tech companies can have on one's life and production is staggering.

I started using Mac OS X Jaguar in December 2002, when I bought my first iBook. I still have that white machine, and it still boots like the first day, even if the battery is utterly dead. Before that I was a Windows XP user, having used (and programmed for) all versions of Windows since 3.1.

The reason I bought that iBook was because I wanted to learn 2 things:

- Objective-C, and
- The Unix command line

That 2002 iBook was not my first Mac. Such honor belongs to a PowerBook 150 I bought in 1994 while studying in the University of Geneva. That little machine (Motorola 68030 CPU, 2 MB RAM, 40 MB hard disk) was a good companion during my studies. But it was not by far my main machine. In any case I did not write any code in it, even though a friend installed a copy of Metrowerks CodeWarrior in it. Sadly, at some point in 1997 it stopped working, and then I got rid of it.

In 2008 I started a career as a full-time iOS and Objective-C developer. Those were the glorious times of Mac OS X Snow Leopard. It was the peak of stability, usability and fun for Mac OS X; we might never see anything like that again. Maximum productivity and zero downtime.

As a developer, Xcode 3 was a joy to use. Objective-C 2.0 was all the rage, with `@property` and `@synthesize` and the whole package. It compiled in milliseconds, executed fast, and one could link it to any C or C++ library out there.

Of course, it went all downhill from there. It started with the trainwreck that was Xcode 4. Sensible souls will remember.

Following that trend, by 2016 the quality of the whole Apple offering was so terrible that I decided to start a migration towards something else. The tipping point was the purchase of the crappiest, most expensive laptop I ever wasted money on: a 13" late 2016 MacBook Pro with a useless TouchBar and the worst keyboard I have ever had the bad luck to type in a laptop. Enough was enough.

So I decided to leave the Apple galaxy. The question was, to go where? How? Leaving macOS (as Mac OS X got renamed in the meantime) meant a huge transfer of data from one galaxy to some other.

Platform lock-down is an actual thing, and it is costly.

And I found out I was not the only one in this path.

After evaluating the standard trinity of options outside Mac OS X, namely Windows, Linux and BSD (NetBSD in particular), I decided that I wanted to use Linux full time. In particular, I settled for Ubuntu. I had already been playing with it in virtual machines and Raspberry Pi boards since 2006, and I knew the system quite well. I felt comfortable enough with it.

(I actually considered Oracle Solaris as well, but for a short while. Like, for thirty seconds.)

One does not simply jump ship just like that. There are three basic variables to take care of in such a migration: Hardware, Software, and Data. In the next sections I will provide details about each.

A Note about Ideology

I want to be very clear: I have not chosen Linux out of a sense of doing “the right thing” or “making the world a better place” or “belonging to a community” or anything like that. I do not think that open source or free software are better or worse than commercial software. I will (maybe) explain this position in a future post.

What I do care more about, if anything, is the format in which I store the documents I create during my computing. If there are formats whose specification is not owned by a corporation, and whose standards are publicly available, I choose them. Otherwise, I go with commercial alternatives, because sometimes you just got to get stuff done.

Of course, I care for a certain quality of the software I use. I experience the same proportion of crappy software in FOSS and COTS. The world is not a better place because of the existence of any of these.

All of this means that my choices of operating system are related to my computing needs and knowledge at a particular time in my life. Windows was OK to begin with in the 1990s; Macs were OK during a certain period of time; and now Linux is also good enough for me. Maybe in a few years I will be using NetBSD for the same reasons.

1. Hardware

This is the part of the trip that took me the longest. I wanted to find a good machine that ran Linux without issues, so I started asking around in Twitter to gather some opinions about the subject.

I really, really, *really* did not want to bother with recompiling kernels, device drivers, or editing `/etc/whatever/conf.rc` following the contradicting advice of a bunch of Stack Overflow answers. Seriously. Thanks, but no thanks.

I wanted to find a good machine with good Linux support. I knew this was possible. I just needed to do some research. And I did find quite a few suitable candidates among the big brands:

- Dell XPS 13”, with 16 GB RAM, with raving reviews online, and sold in Switzerland with US keyboard and Ubuntu pre-installed. (US keyboards are a mandatory requirement for me.)
- Lenovo X1 Carbon, widely praised online as Linux compatible. It actually bundles a dedicated “Linux” compatibility mode in the BIOS settings, and

receives official upgrades from Lenovo. I learnt that it was compatible with Arch Linux and had also got official Ubuntu Certification. And then one day I held one of these in my hands, and are they lightweight! What a lovely construction. And what a great keyboard. I was amazed.

Among my friends online, Fernando Cejas recommended a Lenovo Thinkpad T460, similar to this T470 with support for 32 GB of RAM and two hard disks. Fabrice Truillot started using a ThinkPad around 2018 T480s, also marked as compatible with Linux.

Beyond the big PC makers, I discovered a myriad of independent computer makers specialized in Linux-compatible hardware:

- System76, Purism, and EmperorLinux in the USA.
- Slimbook in Spain.
- And finally TUXEDO Computers from southern Germany (recommended to me by Aral Balkan himself!)

The nice things about these brands is that they are independent computer makers; supporting small businesses is important to me.

And of course, there were those smaller and extremely cheap Linux-based laptops based on the ARM architecture came to the list:

- Pinebook Pro
- Pi-Top

Interesting but a bit underpowered for my taste. Anyway, I kept all of this information in a Joplin note, and evaluated all the options.

Peripherals

Of course I had a bunch of peripherals around my Mac, and I wanted to make sure they were compatible.

For my Logitech wireless mouse, Solaar and Piper provide some functions similar to that of the Logitech Options software.

What about my Logitech R700 wireless presenter? No problem. Logitech C930e webcam? Check. Logitech MX Master 3 mouse? Ready. (Yes, I am a happy Logitech customer.)

Finally I discovered that HP had printer software for Linux compatible with my HP Color LaserJet Pro MFP M277dw printer.

So far, all looked good.

Final Choice

I decided to get a TUXEDO InfinityBook Pro 14 v5 laptop with a 64-bit Intel i7 CPU, 32 GB of RAM, and two SSDs. It came bundled with Ubuntu 18.04.

The TUXEDO people are located not far from the Swiss border, and sell their computers with Swiss chargers and deduct the German VAT automatically from the purchase price. It could not be better. The whole buying experience with them was flawless.

They provide full support, and this even includes a USB stick with the Windows drivers for your machine, in case you might want to have a dual-boot setup. I do not have such configuration, but I think it is a thoughtful touch.

After two years, I can get 5 hours in average of full-time use on a full charge of the battery with lots of apps running. The keyboard of this machine is fantastic. I recently upgraded this machine to Ubuntu 20.04, without absolutely no issue at all.

In the connectivity side, it has a built-in Ethernet port, two USB-A ports, one USB-C port (which can be used for charging the battery, and to connect to their docking station), and a built-in SD card reader. The trackpad is good enough (I seriously do not care that much about it) and everything just works: webcam, built-in mic, etc.

To be honest, the weakest point of the machine are the built-in speakers. They are really quite crappy, but the truth is, I simply do not use them. If you fancy listening to music out loud in your Mac, then you might not like this machine. I personally do not really care. I have my old USB Harman Kardon SoundSticks bought in 2003 and they work like a charm.

In terms of stability and speed, in many ways it feels a lot like using a 2009 Mac with Snow Leopard once again. It is super, super, super stable.

To top it off, I started working at VSHN a few months later, and they got me a fantastic Lenovo ThinkPad X1 for work. So now I have my two preferred Ubuntu machines, one for work, another for personal use.

2. Software

As I mentioned above, I chose Ubuntu as the operating system for this new phase. The main reason is the large (very large) quantity of software available for it. Pretty much every single open source project out there offers a `.deb` package ready to be installed. Ubuntu is really popular. That, in itself, is a big bonus.

Now I needed to find suitable software equivalents for all the things I did on my Mac.

Installing Software

There are a myriad of ways to install software in a Linux machine, beyond the usual `sudo apt install` or `sudo make install`. There is even `umake` which you might not have heard of before. Furthermore, these days most popular

programming languages come with their own package managers: PIP, Conan, Composer, npm, go get, Cargo, Maven, dotnet add, Rocks, Swift Package Manager... as if there were not enough of them.

You could also download a `.deb` package and simply `dpkg -i` it, or double-click on it. Getting used to all of these ways to install software was very important for me.

Actually, there is even Homebrew for Linux now. I tried it for a while but I do not use it anymore.

But then I discovered something new: ready-to-use applications; just download and use them:

- AppImage has become kind of a standard lately. Very popular with FOSS desktop apps; just download the `.appimage` file, then `chmod +x` it, and run it.
- Flatpak is slowly getting traction as an alternative to AppImage.
- Finally, Snaps are an Ubuntu-only solution for the same problem. Very useful and lots of excellent apps are ready to be installed with it.

Note-Taking Apps

Note-taking is a basic component of my computing needs. I usually keep my notes application open at all times, searching and writing notes every time I figure out how to do anything.

I need my note-taking application to sync notes across desktop and mobile platforms, because I like to quickly create notes with audio or images when I am on the go.

And when I say mobile, I do mean both iOS and Android; I used both in the past, and I plan to use them in the foreseeable future. There are things I like about iOS, and then there are things I like about Android.

I had been a paying Evernote user from 2012 to 2014, at which moment I moved to Apple Notes, which had a decent set of functionality for what I needed. It also synchronized stuff reasonably well across iOS and Mac devices. Evernote was bloated, slow, and crashed too much for my taste. Apple Notes crashed, too, but less often, and was simpler and faster.

But Apple Notes had a major drawback: there was no “export” function whatsoever.

So first I had to find the location of notes in Mac OS X (hint: they were stored in `~/Library/Group Containers/group.com.apple.notes/`) and tried to figure out ways to bulk-export all of my data.

It was such a mess, I ended up exporting only the most important notes from my collection. It was enough, and convinced me to never again deal with Apple Notes.

For a short while in 2016 I used Bear, a fantastic Mac application that had the two most features I looked for: Markdown support and PDF export. It did look good, too, but seriously by that time I cared more about functionality than looks. I wanted software that worked.

In my search for the perfect note application I considered these options:

- Simplenote, a bit too simple for my taste.
- Microsoft OneNote, another bloated mess like Evernote.
- Tusk, and Electron-based third party client for Evernote.
- Boostnote: interesting, but it was under heavy development when I tried it. PDF export was not yet available, the iOS apps did not yet sync over Dropbox, and the iPad app had no landscape or iPad Pro support. I think it has gotten better now, but I cannot tell.
- nb: Full CLI, written in Bash, uses Git for sync, support for all Pandoc formats, support for bookmarks and images and docs, one git repo per notebook.
- And finally one day I discovered Joplin by pure chance.

In January 2017 I chose Joplin. It is cross-platform (including a great terminal app), has several sync mechanisms (including Dropbox), and has really good mobile apps. I have been using it every single day for over three years, and I cannot state how happy I am, and how good it is. I use it for everything, all the time.

Here is how I moved all of my notes from Bear into Joplin:

1. Exported notes from Bear in backup format.
2. Opened the resulting `bearbk` file as a zip file and unzipped it.
3. Ran the following script, and at the end of the process, the contents of the text of all notes were stored in a folder named `_export`:

```
#!/usr/bin/env ruby

export = "_export"

Dir.mkdir(export)

Dir.foreach('.') do |item|
  next if item == '.' or item == '..' or File.file?(item) or item == export
  filename = export + '/' + item.sub('.textbundle', '.md')
  contents = File.read(item + '/text.txt')
  File.open(filename, "w") do |f|
    f.write(contents)
  end
end
```

At this point in time, I consider Joplin to be simply perfect for my needs. The only thing that I found flaky in it was the whole encryption thing. All of a

sudden it restarts reencrypting all of your notes and god knows why. Other than that, it is great.

Web Browser

Having been a loyal Safari user since version 1.0, I started using Firefox as my main browser mid-2016. At that time I was using not only iOS devices but also Android devices, and I wanted to use a web browser that had some synchronization feature shared among all of these platforms.

I have used and liked Firefox since 2004. I still do. And of course, thanks to it I could migrate my browsing history and preferred extensions from the Mac to my Linux box without any issues.

By the way, I have an iOS device, and since I upgraded to iOS 14, my main browser is Firefox for iOS. *Finally* we can set a default browser in iOS. It took Apple 12 years to add that feature.

I have also Chrome and Opera installed in my Linux machine, but only for testing websites and web apps. And I will install Edge when it comes out of beta, too. Although all of these use the same rendering engine these days, so it does not really make sense to have them all.

Chrome is the new Internet Explorer.

Programming Languages

The great thing about Linux is that virtually every open source programming language is available for it. I have (of course!) installed the GNUstep toolchain to have access to Objective-C, a language that I like a lot.

But not only that: in the past two years I used, learnt or played with Go, GNU Smalltalk, Common Lisp, FreeBASIC, Free Pascal, Rust, GnuCOBOL, Rexx, and so many more, and so many others will come.

And yes, there is Swift for Linux, if you are into that kind of thing. Not for me though.

Music

This one was rather easy. I started using Spotify in 2013; then switched to Apple Music for a few months in 2015. But it was really bad. It was “click play and wait for hours for the music to start” level of bad, and did not even have all the music I wanted to hear! So I quickly went back to Spotify; and the good thing is that Spotify has a great web app *and* a native client for Linux. Boom.

Spotify is so good, it is hard to describe. I even have Spotify in my LG TV now. It is everywhere.

I also have a rather large MP3 collection, and since I was fed up with iTunes, I started using Musikcube instead. It is a lovely player and it is amazingly good.

I started using it on the Mac, and then continued using on Linux. Simple and good.

Text Editors and IDEs

One of the most valuable things I learnt using Mac OS X was, ironically, the Unix command line; and with that came Vim and GNU Emacs. After the TextMate 2.0 fiasco I started using MacVim a lot (with Janus, then I moved to terminal vim, and by 2017 I became a diehard Microsoft Visual Studio Code user.

The good thing of vim, emacs and Visual Studio Code is that they work perfectly well in Linux. I do not use joe or nano much.

As for serious software development, I needed some IDEs, so I got myself a full JetBrains subscription, and apart from AppCode (understandably only available in macOS) I have access to fantastic software for my work. Their software is among the best and most stable ones I have ever used.

Video

Even when using Mac OS X, I always found VLC much better for video consumption than QuickTime; VLC simply plays anything I throw at it. QuickTime would always complain about some missing codec or whatever.

In the command line world of Linux I discovered youtube-dl to download pretty much anything I found online, ffmpeg to convert between formats, and mplayer to play them back. All works perfectly well. I could even open old videos I had kept from the 1990s, stuff I had not seen in decades.

Funny tip: `mplayer -vo caca -quiet -noborder MovieName.avi` to play a video directly on your terminal. It requires mplayer built `--with-libcaca --with-libdvdnav --with-libdvdread` and the of course the end result is unwatchable. Who cares!

Anyway, I digress. These days I am starting to edit my own videos, and for that I found OBS Studio, which I did not know, plus a lot more software I tried to edit videos:

- Shotcut, based on Qt. I used it for the first VSHN.timer video, and promised myself to never use it again. It crashed all the time.
- OpenShot is much more stable than Shotcut, and the story of the developer behind it is brilliant: C#/.NET developer without experience in Linux, started the project in May 2008... and here it is now. The project includes C++ libraries for video and audio processing. I edited the second VSHN.timer video with it, and it was OK-ish, so I kept looking for other options.
- Flowblade is much more solid than the previous two options, and I will use it for my future VSHN.timer videos.

- DaVinci Resolve is a Hollywood-level thing, apparently, but lots of Ubuntu users have reported issues during installation, and apparently it only works well in CentOS. I have found more installation problems if you are curious. In my case I tried to install it, and I could, but it did not start at all, complaining about the ThinkPad GPU. Whatever.
- Other options I found are Lightworks and KDenlive but I have not tried them yet.

Update, 2021-01-20: Add Pitivi to the list above. Also, I'm not using Flowblade anymore for VSHN.timer, having found KDenlive to be a *much* better alternative. By better I understand: stable, fast, and compatible with Windows.

Update, 2021-07-30: More about video in Linux in this blog.

Video editing is certainly an area where the Mac has a large advantage. If you are a video professional, I would certainly not recommend you to move to Linux.

Image Editors

I am not a visual designer, so my needs for graphics are very limited; just a bit of retouching here and there, and converting images into other formats. In those cases Gimp and Inkscape work perfectly well for me, and they support all of the formats I could dream of using. Krita is another excellent option to Gimp. I am using all of them, depending on what I have to do.

I have configured Gimp to use Photoshop menus and shortcuts which helped a lot at first. But then I ended up learning the actual native shortcuts. Muscle memory takes a little while.

As for the few Pixelmator files I had, I converted them to PSD format (Photoshop), and I could open them in my Linux laptop without problem.

Password Manager

When I started my migration progress there was no 1Password Linux client, so I moved to Enpass instead. They have great mobile and desktop clients for all operating systems, and good browser plugins as well. And it synchronizes over Dropbox.

Update, 2020-11-21: Following a recommendation by Tobias Brunner I have switched to KeePassXC, synchronizing my data with Strongbox over Dropbox.

Communications

Thankfully I had stopped using iMessage with friends and family around 2012; since that time we were all communicating in Telegram. So that was another easy switch. Telegram has both web-based and Linux clients, so I can be notified of messages in all of my devices without issues.

There are Skype and Zoom clients for Linux; no need to change anything there.

Other Apps

To make a long story short, this is the full list of “equivalences” I found for many apps I used in Mac OS X:

Apple applications:

- Notes: Joplin
- Safari: Firefox
- Mail: Thunderbird or Evolution
- Music: Spotify for Linux
- iTunes: Musikcube
- iMessage: Telegram Desktop for Linux
- iWork Suite: LibreOffice and there is even AbiWord in case you have old formats to open, and WordGrinder for a full command-line experience.
- Tasks: Microsoft To-Do
- iBooks: Calibre for Linux
- AirDrop: Resilio Sync
- QuickTime: VLC for playback, and SimpleScreenRecorder for screen recording
- GarageBand: Audacity
- iMovie: Flowblade
- iPhoto: Darktable
- iCloud Drive: Dropbox

Third-party software:

- 1Password: Enpass
- Dash: Zeal
- IDEs: JetBrains
- TeXShop: TeXstudio
- Transmit: FileZilla
- VirtualBox: VirtualBox for Linux
- Slack: Slack for Linux
- Twitter: Corebird
- Deckset: first GitPitch, then Ascidoctor-Reveal.js
- And cool-retro-term instead of iTerm2 but just for the sake of fun :)

Terminal apps that stayed untouched across Mac and Linux:

- zsh, emacs, vim, tmux, irssi, mc, w3m, htop, tig, musikcube, wordgrinder, sc-im, k9s, asciinema, asciidoctor, pandoc, git, git-extras, onefetch, podman, dosemu, xdotool, caddy, bat, v4l2-ctl, gphoto2, pyenv, rbenv, nvm, curl, ag...

By the way, here is a tip which might help when moving from Mac to Linux; install xclip and copy this in your ~/.aliases file:

```
alias pbcopy="xclip -selection clipboard"
alias pbpaste="xclip -selection clipboard -o"
```

I hope I am not missing anything in the list above. Feel free to use this as the basis for your own migration matrix.

3. Data

I was a hardcore user of most Apple services. As a paying iCloud user, I used it for all of my digital life. That was, after all, the whole idea of it. But the user experience was awful. All too often I did lose data because of the failures of iCloud to properly synchronize data between devices made by the same company. Unacceptable, and too often, and for a ridiculously high price.

I moved all of my files from iCloud to Dropbox. I evaluated many other options, but Dropbox had many positive points to it:

- I had already used it since 2008, and it always worked well and fast.
- Best synchronization ever. Seriously unbeaten, particularly after comparing it to iCloud (of course!) but also OneDrive and Box.
- They have great mobile clients; I used the Android and iOS version and I liked them both.
- And last but definitely not least, there is a Dropbox Linux client.

In terms of local storage, Ubuntu is able to access my external hard drives formatted with HFS+ without issue.

Documents

In my life as a conference speaker, I used Keynote extensively from 2006 to 2013. In 2014 I discovered Deckset, which uses Markdown for slides. Of course there is no Deckset in Linux, but I moved my workflow to Asciidoc + Reveal.js slides.

Fortunately, Pages and Numbers documents can be easily converted into Microsoft Word and Excel files, which are completely compatible with LibreOffice, so I did that too.

Well, almost; until a few years ago, the latest version of Apple Pages for Mac or iOS could not open files created in... Apple Pages 1.0 (circa 2005). No comments. Thankfully they added support for that format again at some point. I still can not believe I just wrote this paragraph.

E-Books

After suffering the ordeal of dealing with Apple iBooks, I decided to never buy books from Apple ever again. I only had a few books in “iBooks” format, and of course I will not be able to read them ever again. It is OK, that will teach me a lesson.

I had a few Kindle books, thankfully not a lot; I converted them into EPUB (using Calibre plugins) and coalesced all of my book collection into a great Calibre collection of PDFs and EPUB files.

For the few CHM format books I had (from my times as a .NET developer), xCHM works fine.

I finally bought a Kobo Aura H2O e-book reader in 2016, and since then I have been happily reading books in EPUB format. Calibre recognized it immediately both on the Mac and on Linux. Plug and play.

I also took the decision of not buying any more books for Amazon; I only use their website to find the book I want, and then I buy it directly from the publisher. Most publishers offers EPUB without DRM and in various formats (including PDF and EPUB, and even Kindle .mobi files, although I do not have a Kindle.) The last e-book I bought from Amazon was in 2012.

EPUB files without DRM are the kind of e-book I pay for.

Did Not Happen

As I mentioned previously, I did not migrate to Linux following a false sense of nerdiness, belonging to a community, or some otherwise grandiose feeling of betterment of the world. There are, then, quite a few things that many other Linux users choose to do, that I do not.

Using mutt for mail for example. Or rainbowstream for Twitter, or newsboat for RSS. Or using Org Mode in Emacs.

I sometimes do some command-line web browsing, but for that, I prefer w3m to lynx.

Other things that did not happen:

- Replacing SourceTree with GitKraken or SmartGit.
- Replacing Ulysses with Document Node.
- Replacing Marked with Caret.

Lots of Mac-only software is still unbeatably good. That is just the way it is. For the three last items, I ended up replacing them with Visual Studio Code extensions, or by JetBrains IDEs plugins.

Conclusion

My migration to Linux started in January 2017 and ended in December 2018, when I got my TUXEDO Computers laptop on the mail, and got rid of my last MacBook with TouchBar.

My faithful Logitech webcam, my USB Yeti microphone, and my HP laser printer, all worked out of the box with Linux. And even more ironic, my 2002 Harman Kardon speakers did too. I have two standard monitors that have cost me less than 200 USD each, and they work flawlessly with my two Linux laptops.

I have a productive setup, with a machine with a decent battery life, great specs, all at an unbeatable price (below 2000 USD). My files are saved in standard,

open formats, and I will be able to open them in the future.

During the last two years, this setup has given me much, much less headaches than what macOS was giving me in 2016. It is far from perfect; but it is surprisingly stable, efficient, fast, and oh my god, so much cheaper.

Every user has a different mileage, so I do not advocate migrating to Linux for everyone. Being a developer certainly helped me to find quirks and solve problems here and there. But in general, I had much less issues than I thought I would have. Like, *far* less. I was pleasantly surprised by Linux.

I hope this guide will help you jump to Linux if you feel the curiosity to learn something new. I have learnt so many new things exploring Linux, I am very happy with it so far.

As far as I am concerned, my next step will be to migrate to NetBSD, but not in the immediate future.

PS: I am not interested in receiving opinions about how I could have taken any of the decisions above differently. Please do not contact me to let me know your point of view or to make suggestions. I do not care and the fact that there are no comments in this blog should give you a hint about the fate of any e-mail you might want to send to me.