

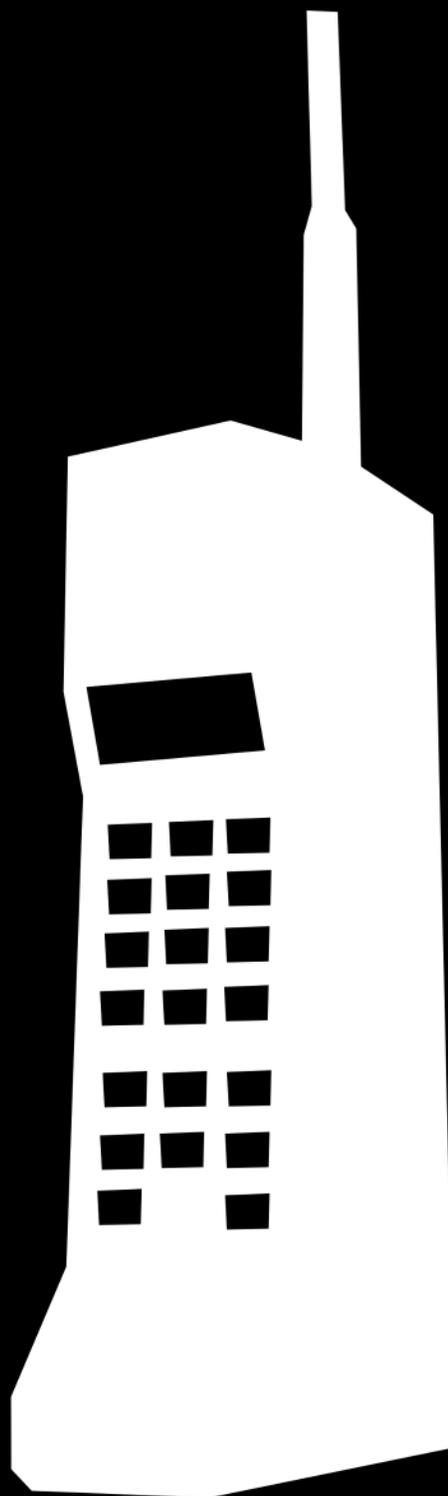
Issue #1  
Aug, 2018

# CAPITAL F

... as in Freedom

## The Road to a Free Software Phone

From the N900  
to Ubuntu Touch  
to the Librem 5



Video Production  
with Free Software

Replacing Google  
with Free Software  
Solutions

# Capital F licensing

because licensing is important

Capital F magazine is licensed under the:

Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

# In this issue...

Capital F – Issue #1 – August, 2018

## The Road to a Free Software Phone

by Bryan Lunduke  
Including an Interview with Todd Weaver  
CEO of Purism

## The Stigma of Charging for Free Software

by Bryan Lunduke

## Video Production with Free Software

by Gardiner Bryant  
aka “The Linux Gamer”

## A Personal Journey with Free Software

by Bryan Lunduke

## Replacing Google with Free Software

by Bryan Lunduke

Advertiser Info

Purism – [www.puri.sm](http://www.puri.sm)

OpenNMS – [www.opennms.com](http://www.opennms.com)



# PRIVACY, SECURITY AND FREEDOM FOR EVERYONE.



GNOME™



coreboot

Librems preloaded with **PureOS** feature the latest and greatest technologies, including **GNOME 3** and the next-generation Wayland display protocol, out of the box. Thanks to this forward-thinking graphics subsystem, you can now enjoy fluid high-framerate videos without tearing, benefit from enhanced security through isolation between graphical software applications, frame-perfect hardware-accelerated animations and UI components rendering, and better power management.

The Librems are **the only modern high-end laptops and phones where you are in control** and have complete visibility into the operating system, all bundled software, and the deeper levels of your computer. Every Librems comes with **free international shipping**, a one year hardware warranty and **lifetime software updates** with PureOS. And our software updates are intended to make your hardware run better over time, not worse.

Purism laptops are the only independently-made, brand new, high-performance laptops on the market specifically meant to pair recent technologies with **our coreboot port** and the potential to have a **neutralized Intel Management Engine**. Purism laptops also feature a **TPM chip** allowing you to secure your operating system and boot process at the hardware level with your own keys.



<https://puri.sm>

# Letter from the Editor

by

Bryan Lunduke



In the Autumn of 2017, Network World (an outlet which I had written articles for over the course of more than half a decade) decided to change their focus. Instead of the Linux, Free Software, Open Source, & privacy focused articles of mine they had been publishing... they now wanted to focus, exclusively, on “Enterprise Technology.”

Containers. Hypervisors. Things with “stack” in the title.

My long-winded diatribes on software Freedom and discussions of various open systems – although some of the most viewed articles – were no longer in line with the revamped vision of the publication.

Not that there’s anything wrong with “Enterprise” stuff, mind you; I worked in marketing for one of the larger “Linux Enterprise” companies for several years. But it just wasn’t what I was excited about.

No. Sorry. That’s a dramatic understatement. The thought of writing another article about some company creating “Innovative Cloud Technology Stack Infrastructure B2B Solutions” made me want to chuck every computer I own through a plate glass window.

So, in November of 2017, I left my cushy, long-term writing gig at Network World and moved on to pastures unknown.

As I looked around, trying to find the perfect publication to write for... I came up short.

There were a handful of truly excellent magazines creating some remarkable and valuable GNU/Linux-oriented content. Magazines I regularly read and enjoy. But I didn’t want to write simply about “Linux”. Operating System reviews (and the like) are incredibly valuable – and other writers were already doing a fantastic job of that.

I wanted to write about Free Software. Open Source. Free Culture (books, music, movies, etc. released under licenses like Creative Commons). Copyright issues. Personal right to privacy and ownership over your own information.

The stuff that directly impacts our Freedoms. With a capital “F.”

Flash forward several months, and here we are. The inaugural issue of Capital F magazine.

This magazine is licensed under the CC BY-SA 4.0 license. Which means, in a nutshell, that you can share it, and modify it, to your hearts content. It’s both free (as in you don’t need to pay for it) and Free (as in you have the right to use it how you see fit... for the most part). See the front page of this issue for details on what this license means.

In order to fund this venture, I've been lucky enough to line up a couple of Freedom-loving companies to advertise on the pages within. You can also purchase physical copies of this issue (and contribute to the production) at the Capital F website ([CapitalF.org](http://CapitalF.org)).

Capital F is bi-monthly, with new issues released every two months. You are reading the August, 2018 issue; Issue #2 will be released (under the same license) in October, 2018.

This magazine is family friendly. Safe for work. And, while many of the topics are decidedly nerdy – an effort is made to be accessible, and enjoyable, for those completely new to the concepts of Free Software and Free Culture.

So give it to friends. Pass it around. Print it out and tape it to your office door. Read it out loud on the street corner. It's yours to do with as you please.

# The Road to a Free Software Phone

by  
Bryan Lunduke



At the time of this writing, the Linux kernel (licensed under the GPL), dominates the mobile operating system space. With nearly 80% of the marketshare, worldwide, the Linux-powered Android is the indisputable king of smart phones.

And yet, here we are in 2018... with no truly viable smart phone powered by predominantly Free Software.

A case could be made that significant parts of Android (AOSP – the Android Open Source Project) are open source and, therefore, we have an Open phone. The reality is far murkier. Significant portions of any common Android installation are closed source and tied to similarly closed source online services run by Google.

Over the years there have been several attempts to build and ship a GNU/Linux-based smart phone operating system, often with design principles and architectures that allowed them to take advantage of the vast catalog of Free Software available on traditional Linux-based desktop systems. Unfortunately, thus far, none have managed to capture a large enough audience – and were structured in such a way, from a business standpoint, where they have not been able to stick around for more than a handful of years.

## Nokia & Maemo

While not a cell phone platform (at least not initially), Nokia released a Linux-based “Internet Tablet” in 2005 called the Nokia 770. “Tablet” is a bit misleading here as it was more similar in form factor to a modern smartphone or a late era PDA – a pocket-sized, handheld device that was mostly screen.

Built on a system known as Maemo, it had most of what desktop users had come to expect from Linux-based systems. A variant of the Debian distribution of GNU/Linux and using a mobile version of GNOME 2, the 770 was capable of running many desktop applications (albeit with a bit of tweaking).

Over the next few years, Nokia would go on to release a few updates to this product line (including the N800 and N810), eventually culminating in 2009 with the N900 – the first true Smart Phone in the series.

I had the good fortune of owning an N900 and consider it to be one of the finest Smart Phones ever made. The gloriousness of having a fully featured terminal, a large repository of software (complete with familiar software management tools), was truly phenomenal at the time. By today’s standards, the N900 is slow and underpowered (with a comparatively low resolution screen). But, back then? It was remarkable; It gave GNU and Linux users hope for a truly Free mobile future.

Unfortunately that hope was short lived. By 2011 the N900 was superseded by the N9 (also Linux / Maemo based) – a fantastic and (at least somewhat) Open phone in it's own right – which was not long for this world. The final update occurred in 2012... and the system quickly became difficult to find in usage.

## **Palm, HP, & webOS**

Around the time of the N900, Palm Inc. (the company behind PalmOS) released a new operating system built on top of the Linux kernel and focused heavily on using “web” technologies for applications. Initially the operating system itself was not Open Source (other than a few components), but the fact that it was using Linux for the kernel attracted many in the Free Software and Open Source communities to the system.

Palm's WebOS first shipped on a smart phone in 2009, followed almost immediately by the company being purchased by Hewlett Packard. A tablet based on WebOS (known as the HP TouchPad) shipped in 2011... and then the system began a rapid spiral into obscurity.

In 2012 a mostly Open Source version of WebOS was shipped (lacking a few components of the closed version of the system). But it was too little, too late to attract much interest from a developer or consumer perspective.

## **Mozilla & Firefox OS**

As WebOS was fading out, Mozilla (the company behind the Firefox web browser) decided to get in on the mobile operating system action.

The idea of Mozilla's Firefox OS was, at a high level, extremely familiar: Linux kernel, applications developed using web technologies. Essentially a web browser being the core of the operating system. And, what's more, the system was predominantly licensed under the Mozilla Public License (a Free and Open Source Software license). A big step in a Freedom-loving direction.

By 2014, things were looking promising. Phones running Firefox OS were available in 28 countries and, while most of the devices were considered to be underpowered and poor performing for the time, the commercial availability (in many markets) suggested a bright future.

As proved to be a trend in Free and Open Source phone systems, things changed quickly.

Later in 2014, Firefox OS added closed source components (such as support for DRM schemes) and, by the end of 2015, cell phone carriers ceased selling phones with the system entirely. Firefox OS was dead.

## **Samsung, Linux Foundation & Tizen**

In 2011, the Tizen Association – run by the Linux Foundation – was launched. Samsung, Panasonic, Sprint, Intel and several other companies joined the association over the years, lending some weight and prominence to the system. Tizen is built, primarily, as an Operating System for Samsung-based devices: their Gear line of smartwatches and their Z series of smart phones.

In many ways, Tizen is incredibly familiar to those familiar with desktop Linux-based systems. At least from an architectural perspective.

Linux kernel. X.org display server. Enlightenment (a desktop environment and SDK that has existed in the Linux world for many years) as the user interface toolkit.

Unfortunately Tizen is notoriously buggy, with the primary company behind it (Samsung) being slow to address bugs (according to developers) and being highlighted as having an unusually high number of security vulnerabilities.

Tizen still exists today. But, despite being free (in cost) for Samsung to utilize, the company opts to continue relying on the operating system of a competitor (Google's Android) for the vast majority of their smart phones. Which is... concerning.

## **Jolla & Sailfish OS**

In 2013, a company named Jolla announced a new smart phone running another Linux-based system known as Sailfish OS. This new phone was a bit more competitive, hardware wise, with iOS and Android smartphones of the time and generated quite a bit of excitement.

Despite being based on a Free Software kernel, and utilizing Mer (a Free Software base distribution) Sailfish OS itself was not entirely Open Source. The company stated that they intended to release source code for the full system, but that has not yet fully materialized (much of the user interface is closed source).

Availability for the Jolla phone is limited (at the time of this writing I can't find a way to purchase one), but the operating system itself has been ported to a number of different smart phones.

## **Canonical & Ubuntu Touch**

In 2013, Canonical (the company behind the Linux-based Ubuntu) released the first developer version of "Ubuntu Touch" – a touch-centric (as the name suggests) smart phone version of their desktop operating system.

Though the system was plagued with stability and performance issues, the fact that it was (primarily) Free Software – using a nearly identical architecture to its desktop counterpart – held a great deal of promise. It even became possible to run many desktop applications (such as the LibreOffice office suite) on Ubuntu Touch devices. Albeit, rather slowly.

The first Ubuntu Touch powered device shipped in 2015, with a tablet following in 2016. Updates, after this point, slowed considerably. In early 2017, the company announced that the platform was being discontinued, development was ended, and support servers would shortly go offline with only a few months notice for customers.

## Purism & Librem 5

In 2017, Purism, a company that (up until this point) had focused on laptops running GNU/Linux, announced plans to build a smartphone that was focused on Free Software and privacy (including hardware kill switches for networking, microphone, and camera).

The company launched a crowd-funding campaign that brought in over \$2.5 million, with the stated goal of shipping in early 2019.

[Full disclosure: Purism took out an advertisement, for their line of laptops, in this issue of Capital F.]

### Where We're At

The road to this point has been... let's just say "rocky." We are nearly a decade after the launch of Nokia's N900 with no currently shipping, sustainable, Free Software phone that average people can purchase and use today.

With the promise of the upcoming Librem 5 – and the high hopes pinned on it by many – I sat down with Todd Weaver, the founder of the company behind it, to pick his brain and get a better understanding of how they're approaching building a Free Software phone.

What follows is the unedited interview. We started with some light questions, to get a better feel for Purism's ethos and goals, and then get into specifics around how they're building the Librem 5 – and how they feel it'll succeed where so many others have failed.

**Bryan**

Why does Purism do what Purism does?

**Todd**

To steal one of your great lines and adapt that to why we do what we do, is because "We want to treat people awesome." And the current technology giants, and the trends of technological choices are steering society toward a dystopian future. So at our core, we believe people should have their digital rights protected. What that translates into is a radiating series of decisions to help with that core goal; such as forming Purism as a Social Purposes Corporation, solving the issues of security by default, privacy protection by default, and of course utilizing free software everywhere possible as that is the nucleus of all smart technological decisions.

Having this belief as our clear marching orders, means we are continuing every moment to advance that mission. Behind the scenes it is really wonderful to see a team of dedicated individuals discussing how to best implement the reason why Purism does what we do; from hardware selection negotiations, education of the supply chain about releasing

drivers, releasing free software ourselves, involving the community in our projects, selecting default applications to help users, and generally caring about human beings and their digital life.

We are working daily to create a future of computing that is an ideal place to live. A utopian digital society.

**Bryan**

In working towards that goal -- that utopian digital society – what do you find to be the biggest challenge (technical or otherwise)?

**Todd**

It comes down to two words, convenience and control. It is convenient for people to give up control, incrementally and over time, this amounts to your digital rights being violated, which we are experiencing today.

To participate in digital society you use convenient products, and you will give up control for convenience.

So how do we as a caring awesome group of digital rights fighting people combat this control from the unethical big-tech corporations? By making products that respect people that are convenient to use.

If we make products that are as convenient, or more convenient, than the proprietary rights stripping alternatives while allowing people retain control of their personal lives and still participate in digital society, we--as people--win.

Regarding challenges, Purism has overcome some of the largest challenges already, proving we can deliver convenient products, proving that we have a business model that can both create secure products and respect the digital rights of people, and proving that there is a market demand for these things both individually and at the enterprise level. As we grow, we gain leverage to change the future for the better, and that is a good thing.

With the biggest risks removed, we are now looking ahead toward growth management and Librem 5 phone development. Growth management being investment into enterprise sales, promotion and education, additional research and development, and staffing. Librem 5 phone development is of course our upcoming product that is prime to disrupt the 500 Billion dollar a year phone market by providing an alternative to the Google Android and Apple iOS duopoly.

While there are always challenges, I believe Purism is the best answer to

overcome those challenges, and advance our digital world toward something I would like to live in.

**Bryan**

Let's focus on the Librem 5 for a moment. There have been past attempts at bringing "real Linux" to smart phones -- Maemo, Tizen, and Ubuntu Touch spring to mind. If the track record of past companies attempting this is any indication... this is a tough nut to crack. How is the Purism approach different from past efforts?

**Todd**

A few in bullet form:

- \* Purism does hardware + software
- \* We have a record of delivering
- \* It will be our flagship device
- \* We are working with upstream projects
- \* We are also addressing the security market
- \* We are hyper-focused on only 5 apps for delivery
- \* We are all aligned on a common goal internally

**Bryan**

What are those 5 apps -- and why are those the apps you chose to focus on?

**Todd**

1. Phone calls
2. Text messaging
3. Encrypted messaging (Matrix)
4. Web browser
5. Encrypted email

Those apps get us to the point of offering a "full communication device", so by focusing on the most important category of a phone --communication--we created the smallest list we knew we could deliver on, while still having it offer strong functionality. Apps will increase in number, especially after delivering 500 or so developer kits in June 2018 along with helpful developer documentation <https://developer.purism.org> to create more applications before January 2019 phone shipments.

**Bryan**

The inclusion of Matrix as a default encrypted messaging platform is an interesting one. This would make Purism, to my knowledge, the only Phone manufacturer (or Phone OS developer) to include a decentralized message network -- one that no single company can totally control -- by default.

Giving up that much control (centralization) would give Apple and Google nightmares.

Along those lines, you've also talked about the ability to install other GNU/Linux systems on the Librem 5 -- this seems to be the ultimate in "not locked in" hardware. Are you encountering any difficulty in taking this approach?

## **Todd**

Correct, Purism's top-level approach is to follow a pretty simple set of guidelines; encryption, decentralization, free software, no tracking, and convenient. So the decentralized, free software, encryption nature of matrix fits nicely toward that goal.

Our statement of any GNU based OS comes with one asterisk, we are pushing upstream to Debian, GNOME, KDE, and projects, and PureOS will gain adoption of our advancements first. So if day 1 you are expecting a Debian OS you may want to consider that it will probably be a few months behind on what we're doing with PureOS. And of course any downstream of Debian, or other GNU OSes have to pull all this together.

As long as that is understood, you will also see we have not encountered any difficulty. Any difficulty we may see will be around policy of the other distributions to adopt our approach, such as "do we want to include x app, or y app in our distribution like what PureOS is doing."

Our app store, which of course is just a repository available to install applications from, will only allow apps with FSF approved free software licenses, so will also be available for other distributions besides PureOS.

## **Bryan**

Speaking of "FSF approved"... Richard Stallman and the Free Software Foundation are notoriously strict about what they'll put their stamp of approval on. Do you think it's possible to, one day in the not-too-distant future, get to a point where the FSF would grant their official approval to a smart phone?

## **Todd**

The FSF has a very high standard, with clear reasons and rationale, I am a huge supporter of their policies, since those policies have proven to be accurate litmus tests of protections for users time and time again.

The Purism Librem 5 has the best chance of gaining the Respects Your Freedom (RYF) official approval from the FSF, since we are designing the hardware with RYF in mind. We are in semi-regular communication with FSF about our progress and any areas that need clarification, we will be sharing our dev kits, and early production run hardware with FSF, since we believe we will comply with the RYF standard for the Librem 5 phone. That hardware coupled with the already FSF endorsed GNU distribution PureOS, will make for an ideal offering.

# The Stigma of Charging for Free Software

by  
Bryan Lunduke



No ifs, ands, or buts about it – when people (and companies) charge a fee for Free and Open Source Software... people get cranky.

Take the example of elementary, a GNU/Linux based operating system. A few years back they changed their download page to encourage people to donate to the project prior to downloading their software. In fact they made “Donate \$X” the default option, requiring people who wished to download without paying to perform an extra click of the mouse.

The response from certain corners of the Internet was downright persnickety. Ranging from mildly critical to outright hostile (and not exactly family-friendly):

“ElementaryOS can go kindly \*\*\*\* itself”

“unlike most free software projects they aren't doing it for a love of what they are creating or to share”

“Passive aggressive bs”

And that was some of the more family friendly quotes.

Canonical, the company behind Ubuntu, did something very similar with their GNU/Linux distribution. Want to download Ubuntu without donating monetarily? You'll need to click an extra button. The Internet responded with a similar ferocity.

Neither elementary nor Canonical were charging for their wares – simply requiring an extra tap of the index finger in order to download for free. A way to encourage people to think about donating while grabbing the download. Despite this, the anger expressed bordered on the extreme.

This same phenomenon holds true for other mechanisms of monetizing (read: “selling”) Free and Open Source Software – as well as other forms of content released under Free Culture licenses (such as books or videos released under a Creative Commons license). Pay-to-download being the most common. There exists a subset of the “Free Culture” supporters of the world that become downright livid at the notion of paying for something that is Free (capital F).

I've witnessed this with my own work. Not long ago I began releasing the shows I produce (audio and video) via a subscription service. I moved away from a non-Freedom-respecting service (YouTube) to a system where all of my shows are provided DRM-Free and under a Creative Commons license.

Despite going from “less Freedom respecting” to “extremely Freedom respecting”... the response from some was, to put it mildly, a bit grumpy. That grumpiness being focused entirely on the “pay money” aspect.

While I understand the negative feelings that many have around “paying for something Free”, those thoughts are at odds with even the Free Software Foundation and Richard Stallman himself.

The following quotes are from an write-up entitled simply “Selling Free Software” from the GNU project (started by Richard Stallman):

“Many people believe that the spirit of the GNU Project is that you should not charge money for distributing copies of software, or that you should charge as little as possible—just enough to cover the cost. This is a misunderstanding.”

“we encourage people who redistribute free software to charge as much as they wish or can”

“The word “free” has two legitimate general meanings; it can refer either to freedom or to price. When we speak of “free software”, we're talking about freedom, not price. (Think of “free speech”, not “free beer”.) Specifically, it means that a user is free to run the program, study and change the program, and redistribute the program with or without changes.

Free programs are sometimes distributed gratis, and sometimes for a substantial price. Often the same program is available in both ways from different places. The program is free regardless of the price, because users have freedom in using it.

Nonfree programs are usually sold for a high price, but sometimes a store will give you a copy at no charge. That doesn't make it free software, though. Price or no price, the program is nonfree because its users are denied freedom.

Since free software is not a matter of price, a low price doesn't make the software free, or even closer to free. So if you are redistributing copies of free software, you might as well charge a substantial fee and make some money. Redistributing free software is a good and legitimate activity; if you do it, you might as well make a profit from it.”

In short:

“Selling a copy of a free program is legitimate, and we encourage it.”

This is the stance of GNU, The Free Software Foundation, and Richard Stallman. It is mine as well.

While I respect and understand the frustration some people feel when they see “Free Software” (and other forms of “Free” content) being provided for a price... I would posit that those vocally opposing charging for Free Software are doing considerable harm to not just those creating said software, but to the entire Free Culture movement as a whole.

I won't speak for those individuals. Each of them may have very different motivations for expressing their anger at “paid but Free” software and other content. Perhaps they feel that all things (no matter the form) should be free of any cost. It could be that they hold to a modified form of what the Free Software Foundation promotes – where items released under a GPL (or similar) license must not be

restricted by fees in any way. Or, perhaps, they're on a tight budget and the notion of paying for something they feel should be free makes them grumpy.

There could be a thousand reasons for their opposition to paid-but-Free. Some may be very relatable and understandable. Regardless, that opposition serves to divide and demoralize the world-wide community of those actively producing and promoting Freedom-respecting software, books, videos and other content of value.

# Video Production with Free Software

by

Gardiner Bryant, aka “The Linux Gamer”



When I first started editing video, I was using rudimentary tools. Back then I was a young lad, no more than 14 years old. My “editing rig,” if you can even call it that, was a Windows XP machine and the only software I had at my disposal was Windows Movie Maker.

And my camera setup was no less primitive. I was using a Sony camcorder that was about 6 years old at that point.

I mention all this because I think it taught me an invaluable lesson; What you are making takes precedence over how you’re making it.

I would splice together angsty music montages that told a story visually, narrative short films with my church group about the apocalypse, and short comedy sketches with my friends.

I became known as ‘the video editing guy.’ I was asked by lots of people to make videos for them. All the while using Windows Movie Maker.

Flash forward to last year and a film shot on a smartphone was nominated for an Oscar.

I believe the lesson stands to this day: how you make something is irrelevant if your content sucks.

There’s this idea that video production on GNU/Linux is terrible. (And I mean, you can switch out video production for anything and it would probably be a widely-held belief among non-Linux users.)

But the fact is, it’s not. Sure, KdenLive isn’t at feature parity with Adobe Premiere, and Natron might not be a drop-in replacement for After Effects.

But if you’re willing to give FOSS solutions a try and you have a modicum of talent, I can guarantee you’ll be able to achieve works of art.

Let’s talk about some of the software I use:

By far the most powerful non-linear video editing software is KdenLive. I discovered KdenLive through Jordan Keyes’ This Week In Linux. I was shocked by the production quality and the fact that he was doing his editing in FOSS software. It had a profound impact upon me. It’s pretty strange to acknowledge that I’ve been using Kdenlive for over eight years...

I was impressed back then how great KdenLive actually is (and I remain impressed to this day). You have multiple video and audio tracks you can work with. Compositing various elements together is a

breeze. Proxy clip editing is a lifesaver. Rendering is rather efficient and has an array of container and codec options. And projects can have a mix of formats, frame rates, and compression schemes.

KdenLive is not without its shortcomings though. It would be generous to describe some releases as merely unstable, there can be uneven support for different video container formats and codecs, and it can seem stubbornly single-threaded at times.

With all that being said, following some of my favorite YouTube creators on Twitter has made me realize that this isn't exclusive to FOSS offerings. All these problems are shared with many proprietary editors. So really, that seems par for the course in my book.

OBS Studio is another fantastic tool that I use in nearly every video. It's an open source streaming and screen capturing suite that's a rather recent addition to my Linux video editing arsenal.

It's efficient, stable (most of the time), and flexible. Video recording is a breeze to set up, CPU overhead is minimal (especially when Nvidia's NVENC is enabled), and the footage that generates looks great!

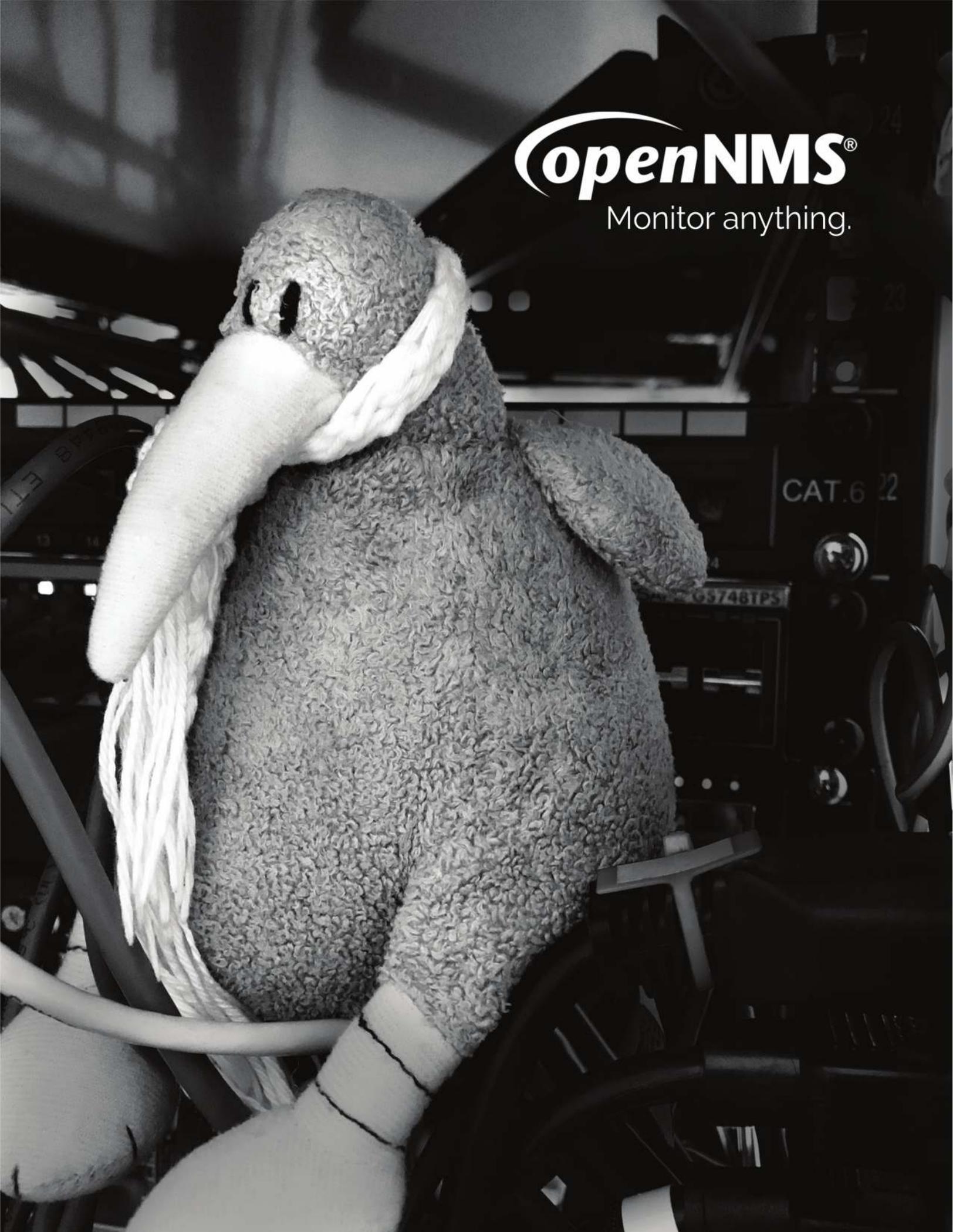
I don't have many complaints about OBS Studio. The biggest one is that OBS and PulseAudio don't play nicely together.

A great example would be when I am running OBS and KdenLive at the same time (which is something I do when I'm streaming my video editing process on twitch.tv/xondak). In this scenario, the sample rate for my USB mixing board seems to change automatically and my voice sounds like it's lower than it actually is. It sounds like I've inhaled sulfur hexafluoride. It's scientific!

In summary, I find that video production on Linux, while perhaps having fewer options, is a valid option for content creation. I love the software that I use and I believe that we, as the FOSS community, are as capable as ever.

**OpenNMS<sup>®</sup>**

Monitor anything.



# De-Googling your life with Free Software

by  
Bryan Lunduke



Back in the summer of 2015 – almost exactly three years ago – I wrote a series of articles entitled “Kicking Google out of my life.” That endeavor turned into an 8 part series chronicling my efforts to eradicate every trace of Google (and Google related service) from my computing experience.

And it was not easy.

Over the few years leading up to the summer of 2015, Google had become a cornerstone of my computing life. For email I used GMail. For most writing I used Google Docs. Android, Chrome, ChromeOS, Google’s Play Store... it was all part of my daily living.

I knew, all along, that if anything happened to Google, I’d be toast. All of my eggs were in the Google basket. This was, on my part, profoundly dumb. Yet I allowed it to be so.

Hence, the “Kicking Google out of my life” goal I imposed upon myself. I knew a change had to be made, and I knew it would be at least a little bit painful – as such change so often is. Three years later, and my life is about as Google-free as you can get – with the exception of YouTube (a service I am using less and less with each passing day).

At this point I can’t imagine seeing a need to head back to the warm embrace of Google.

What I’d like to do now is document, for all of us, my personal recommendations for how to replace some of the more high-profile Google products and services with alternatives that not only respect your Freedoms... but provide you with additional flexibility, security, and control over your own data.

Let’s start with the obvious one...

## Google Search

There are many, many, alternatives to good old “Google” when it comes to searching. I’m going to present three that I find the most interesting here. However, don’t consider this a complete list. I encourage everyone to hunt around and find a good search engine that works well for you – one that, preferably, does not track your searches (like Google does).

DuckDuckGo (duckduckgo.com) is a mix of both proprietary and Free Software. While I would love to see the remaining closed source code within DuckDuckGo become Free (or at least Open Source under a variety of licenses), I admire their “no tracking users” stance. And, I’m not the only one. In 2016 the Tor Browser (a modified version of Mozilla’s Firefox that utilizes the Tor anonymizing

network to make some online activity more difficult to track) made DuckDuckGo their default search engine.

Plus, using DuckDuckGo comes with a good alternative verb for “searching online.” Instead of saying “Google it”, you can now say “Duck it.” Has a nice ring to it.

Two other options that do not track your activity are StartPage (startpage.com) and Disconnect (search.disconnect.me). Both have been highly recommended by those who are security and privacy conscious. Though I tend to stick with “Ducking” things, myself.

## Google Mail

Kicking the GMail habit can be a bit tricky. Once you get used to a free (that you don’t need to pay for) email service with huge amounts of storage... it’s hard to leave! Plus, and let’s be honest here, GMail is pretty reliable; It’s rare that GMail goes offline for any real length of time.

Luckily, if you are set on moving to a different email service, there are some benefits. I’m going to focus on two of the best options (as I see them).

ProtonMail (protonmail.com) is probably one of (if not the) most secure email providers out there. What’s more, they offer a free (in cost) option. If you need more features (like being able to use your own domain for email) they have paid upgrades.

There are some downsides to ProtonMail.

For one, you must use either their web interface... or their provided mobile applications. ProtonMail does not support standard email interfaces such as POP or IMAP. Like using a particular desktop email client? You’re plum out of luck there. And, while the web interface for ProtonMail is open source under the MIT license, their mobile applications (such as for Android or iOS) are closed source and proprietary.

I, personally, utilize KolabNow (kolabnow.com). Unlike ProtonMail it is, very much, a traditional email service – with POP and IMAP support. The company is hosted in Switzerland and, importantly, the entire source code for the service is available. KolabNow makes it pretty easy (at least as easy as it can be) to download your email database from their server and setup / run your own instance locally.

Which means, should you ever want to leave KolabNow, you have an easy exit strategy. I absolutely love that. It is a paid service, with no free option, but I find the value of having added data freedom – plus additional privacy of not having Google know what is in my emails – makes it well worth it.

## Google Drive & Docs

Storing most of our files “on-line” has become the norm. And, while I am a big advocate of effective, local back systems – I can’t deny the extraordinary handiness of online backup and file sharing. Makes it incredibly easy to sync files between machines and collaborate with others. It’s almost become a “can’t live without” piece of functionality in modern computing.

And, luckily, if you're ready to break free of Google Drive (or similarly closed services from the likes of Microsoft or DropBox), there is a truly excellent option:

Nextcloud ([nextcloud.com](http://nextcloud.com)).

It's Free and Open Source Software that runs on a server (you can host it yourself, or pay for one of the many Nextcloud hosting options out there) and handles pretty much everything you could want from a file backup and synchronization system. There are even native clients for the major desktop operating systems (including Linux, something Google Drive lacks) and mobile OS's (such as Android & iOS).

Nextcloud also has many "applications" that can be loaded into the web interface – adding functionality like calendaring, collaborative document editing (such as you get with Google Docs), contacts, and a bunch of other handy things.

I've been using Nextcloud since it was originally released (Nextcloud is a fork of OwnCloud... and was created by the founder of OwnCloud) and have never, not even once, encountered any significant issues in using the system.

It respects my Freedom. It allows me to control the security and privacy of my files. Really couldn't ask for much more than that.

## Google Chrome

This is an easy one. Just use a different web browser. Any web browser. Firefox. Brave. Vivaldi. All of them work well enough to get your browsing fix in... in a decidedly non-Google way.

Personally, I recommend Brave ([brave.com](http://brave.com)). Not only is it Free and Open Source, but it comes built in with excellent tools to automatically block tracking scripts and advertisements. This not only helps to increase your security and privacy online, but it has the added benefit of making many websites load significantly faster.

Brave even has a start page that shows you how many tracking scripts have been disabled for you (on that machine) to date... along with an estimate of how much load time has been saved. A nice touch.

## Android

Up until this point, replacing Google services has been relatively easy. In fact, once you replace GMail, Google Drive, etc. with some of these more Free options... you actually gain some significant advantages.

Replacing Android, the OS that is (in all likelihood) powering your phone or tablet, is a bit trickier.

The key issue here is that most phone hardware makes it *very* difficult to replace the operating system. Thus... you end up stuck with Android.

Luckily there are other options, so long as you are willing and able to buy new phone hardware. See the article on Free Software Phones in this issue for some of the options available.

# A Personal Journey with Free Software

by  
Bryan Lunduke



I started my software career in the 1990s at Hewlett Packard, followed by several years at Microsoft.

In those days my focus was on Windows, OS/2, HP/UX, and MacOS (the “classic” type, mostly... back before they added an “X”). Sometimes Solaris (this was long before Open Solaris was a thing).

The systems I worked on? Closed source. All of them. 100%.

The code I tested and wrote? Also closed. Every last byte.

This was fine. At least that’s how I felt at the time. I never worried about the license of the software I was using or helping to build (other than in the “how many people are allowed to use this at once” sort of way).

It just wasn’t an issue on my radar. Source code for most software was unavailable in the same way that water is wet and donuts are delicious. Eternal, undeniable truths.

Then, one day, I picked up a small stack of early 1980’s computer magazines from a used book store (in this case it was BYTE magazine). Because I’m a nerd. And they looked awesome. I had read BYTE (and a few others) when I was a kid, back when they were new. The nostalgia of it all was irresistible.

As I flipped through – amused at advertisements for hard drives that cost more than my car (and were probably about as heavy) yet only held as much as a few floppies – I started looking at the code sections.

Several pages, in various parts of the magazine, were simply code listings. In an early 1980’s form of the BASIC programming language. Line numbers. Line numbers everywhere.

It felt weird. Not because of the line numbers (though that did give me a good chuckle as I reminisced about my days on C64’s and Apple II’s). What threw me off was the fact that I was looking at code in a magazine... *at all*. By this point in the 1990s, you really didn’t see that so much anymore.

Occasionally in a PC magazine you’d get snippets of Batch files (ask your parents). And in Macintosh magazines (yeah, we still called them “Macintosh” back then) you might stumble across a teensy, tiny bit of AppleScript. But that was usually the extent of it.

Not in this 1980s copy of BYTE. There were whole applications here. A few of them. And a game. Pages upon pages of BASIC code, ready to be typed into a TRS-80 (or similar system). It was *beautiful*.

On a whim I fired up a copy of Microsoft QuickBasic (yeah, yeah, make fun of me later). Started typing them in. Would it work? Would these old pieces of software, created for a completely different type of computer, decades earlier, run on a modern PC?

You know what? They did! At least for the most part. I had to make a few tweaks here and there where this particular BASIC syntax differed. But it was pretty trivial changes.

The resulting software was primitive by the current standard of the day. No graphical interface. Very few features. Almost zero “error checking.” But, hey. They worked. I had recreated, from simple instructions, a computing experience that had been discontinued and abandoned by its creators years earlier. Eons in the computer world.

I felt, for however brief a moment, like the Indiana Jones of software. I was peering into, and recreating, a glorious relic of the past. And it took me barely an hour.

After that I started noticing how rarely I saw actual source code in the world (other than for the software I was working on, myself). Not in magazines. Not as part of the operating systems or software I purchased. It started to bug me.

Not because I felt like I deserved the source code. Not because I thought developers had a moral obligation to provide their code to me or anyone else – code that they poured blood, sweat, and tears into.

But... what if... that code was lost? What if it was backed up to single floppy disk, and that disk failed? That piece of computing history, however small, would be gone. Blown away and mingled with the sands of time.

And so, my first foray into “Open Source” truly began; Because I hated the idea of losing a part of computing history.

As I dabbled with my early expeditions into Linux-based systems (predominantly SUSE, Red Hat and Mandrake) my Open Source fascination grew. With each year I came to appreciate, at an ever increasing rate, the benefits of having such widely available source code – beyond the simple benefit of historical preservation. I was hooked.

By the time 2006 rolls around, I’d become a vocal advocate for Open Source (and, to a lesser extent, Free Software). I wouldn’t call myself “hardcore” about Free Software... I just rather liked it a fair bit. Though, boy howdy, did I have my disagreements with the more intense Free Software advocates out there – including Richard Stallman, himself.

Then, in 2015, something clicked. What follows is an article I originally published in Network World that covers my mental shift towards becoming, as I put it at the time, a bit of a Free Software extremist.

### **The World We Leave Our Children: How I became a free software extremist**

How the birth of my second child changed the way I feel about Free and Open

Source software and privacy.

Originally published September 24, 2015

On September 8th, my son was born. Being my second child, I was far less... let's call it "completely freaked out and scared that everything was going to go wrong and I was going to accidentally kill my baby." I wasn't exactly "super chill" about the event, but far more so than with my first child.

Relaxed enough that, as I held the little guy in my arms that night – mom sound asleep – my mind filled with thoughts of the future. Of the world that I leave to both him and his big sister.

Most days I find myself focused on immediate and practical matters. Making sure there is a roof over their heads, food in their bellies and knowledge in their brains. But, on that night, my mind found its way to a far more philosophical place.

Not so much, "how do I prepare this child for the world that exists?" as "what do I want the world to be like for him?"

A little digression...

Over the years, I've had countless debates – with some of the greatest minds in the Free and Open Source Software world – about how to practically achieve a higher level of Open Source within a world that is so often... closed and proprietary. How do we encourage companies to add a little Open Source here and there? How do we find mechanisms to fund small amounts of Free Software in an otherwise largely closed source ecosystem?

These have been the questions that I've asked. This has been much of my focus.

And, almost without fail, the great proponents of Free Software have dug in their heels when I present such questions. "No," they say! "There is simply no room for half measures and compromises! The source code is either Free and Open or it is unethical and cannot be tolerated!"

To which, again almost without fail, I become indignant and declare that stance to be far too hardcore and extremist to be practical.

These conversations (and, in some cases, heated arguments) have been interesting. But, in the end, I remained staunchly set in the "let's find a practical solution to have a little more Free Software" mind-set. And the Free Software Extremists remained... extreme. Unmovable. I think both sides found that rather frustrating.

Ok. Back to the night he was born...

As I held him, I found myself thinking back on my own struggles in making my own source code... Free. Thinking on a world divided between freedom and restriction. Between openness of technology and a world of litigiously enforced software patents. Between total control over your personal data... and a complete lack of personal privacy.

That core struggle – between Free and Not Free – is a purely artificial one. It is a struggle that we, as humans, created for ourselves.

Source code is inherently "open"... until you close it.

Your personal information is inherently your own... until you grant others the right to take it away from you.

We are free. Until someone comes along and takes that freedom away. Or, perhaps even worse, we give it away. Willingly. In exchange for a shiny new app for our shiny new smartphone – or the illusion that we must give up that freedom for the sake of our current economic system and security.

It was at that moment, as I looked down at my sleeping son's face, that I realized something about myself.

In my mind there was no longer any excuse to keep things closed. To keep things locked down. To give away rights to our personal data. Not even a little bit. I simply would not allow a world that would treat my son and daughter in such a despicable, horrible way. A world that would force him to either give up his freedom or fight, tooth and nail, to try to reclaim freedoms that he should never have lost in the first place.

I had, somewhere along the line, become a Free Software extremist. Just like all those extremists that I'd spent so many years arguing and debating with. It just took two kids to get me there.

Quite a big shift from my early professional years – where I didn't even give software licensing and source availability a second thought.

Where my views of Software Freedom and licensing goes from here, I couldn't even begin to guess. But I find looking back on my evolution to this point fairly fascinating.

# THE LUNDUKE SHOW

DRM-FREE DOWNLOADS - MP4 VIDEO & MP3 AUDIO

RSS PODCAST FEEDS - CREATIVE COMMONS LICENSED

LINUX STUFF - NERD STUFF - RETRO STUFF

[WWW.LUNDUKE.COM](http://WWW.LUNDUKE.COM)  
[WWW.PATREON.COM/BRYANLUNDUKE](http://WWW.PATREON.COM/BRYANLUNDUKE)

# Source Code!

by: various

Long, long ago in a decade far, far away... computer magazines had source code in them. And it was wonderful.

Code that could be typed into your computer and used. Sometimes they were little utilities. Sometimes games. But, regardless, they provided an opportunity to learn something new. With that in mind, Capital F magazine has a source code section. The way it should be.

Below is a BASH script written by a gentleman by the name of Brendan Gregg. It's a simple number guessing game. What's great, is this simple script (comprising a mere 20 lines of functioning code), shows the basics of displaying text, inputing text, doing some math and reading/writing to a text file.

Type this out using your favorite text editor, save it as `"guess.bash"` and run it with a simple `"bash guess.bash"`.

```
#!/usr/bin/bash
# https://github.com/brendangregg/GuessingGame
# guess.bash - guessing game in BASH (Bourne Again Shell)
# 17-Oct-2004 Brendan Gregg Created this.

scorefile="highscores_bash"
guess=-1
typeset -i num=0

echo -e "guess.bash - Guess a number between 1 and 100\n"

### Generate random number
(( answer = RANDOM % 100 + 1 ))

### Play game
while (( guess != answer )); do
    num=num+1
    read -p "Enter guess $num: " guess
    if (( guess < answer )); then
        echo "Higher..."
    elif (( guess > answer )); then
        echo "Lower..."
    fi
done
echo -e "Correct! That took $num guesses.\n"

### Save high score
read -p "Please enter your name: " name
echo $name $num >> $scorefile

### Print high scores
echo -e "\nPrevious high scores,"
cat $scorefile
```

