Att utveckla fri programvara för läsare. Några frågor till Kovid Goyal

calibre är ett program för att samla och organisera e-böcker. calibre är också fri programvara, vilket innebär att källkoden är fritt tillgänglig. Martin Persson ställde några frågor till Kovid Goyal, programmets skapare, om e-böcker, DRM, vad det innebär att utveckla fri programvara i dialog med användarna och hur vi tar makten över vårt datoranvändande och våra data i molntjänsternas tid.

Text: Martin Persson
Illustrationer: Balsam Karam

Hi Kovid! To begin with, could you tell me a little bit about yourself? Where do you live, and how do you spend your days? What's your background, and how did you become a free software developer?

I live in Mumbai, India. I currently work full time on calibre (have done so for the last six years). I have a Ph.D. in Quantum Information Theory from Caltech. As for free software, I have been using Linux as my primary operating system for almost two decades. I started using Linux as a way to learn more about computers (which have always been a hobby of mine). Nowadays, I love open source software because of the extensive opportunities it affords for customization and powerful automation.

calibre offers a rich toolbox for collecting, organising and sharing a library of digital documents, together with functionality for creating and converting ebooks in/between different formats. It started off as a way for you to handle your own personal library and ereader device. What would you say about the current state of ebooks, do you see any differences technologically between the time when you launched calibre and now?

Not really. A few things have improved incrementally, obviously, but qualitatively they are mostly the same. The single biggest change is probably that all non-Amazon ebook vendors as well as all publishers have ”standardised” on EPUB. I put standardised in quotes because EPUB is far from optimal as a standard. EPUB inherits all the problems of HTML in the era before browsers all decided to standardise. EPUB reading software is completely fragmented.

calibre is released under the free software license GNU GPL. What role has free software played for you as a software developer?

In my opinion, free software is by far the most efficient model for the creation of software. Being able to read and modify the source code of software you use/depend upon is invaluable and allows you to leverage the efforts of a global community of developers.

Free software development is to a large extent community-driven, with a network not only of developers but also of users, who file bugs, make feature requests, help out with documentation and support, etc. Has the user community been important for calibre?

Yes, very much so. calibre would not exist without its user community. Bug reports from users are the reason calibre is of such high quality. The development of new features is mostly driven by requests from users, thus ensuring that calibre does things that are actually useful/important to its users. The calibre user community is a very friendly place full of volunteers who help fellow users with many issues related to ebooks and calibre, see the forums at mobileread.com. Many calibre users went on to become calibre developers and have contributed significant functionality to calibre over the years. And last, but not least, the calibre user community directly supports my efforts to develop calibre through donations. Their generosity has allowed me to work full time on calibre for over six years and counting.

So, your work is enabled by crowdfunding. Has this economic basis been stable over the years? And what is the key to this success do you think? It is my impression that even succesful and widely-used FOSS[1] programs can have difficulties in paying developer resources through user donations.

Yes, it has been very stable. There are several things the typical FOSS program gets wrong, that prevent it from being successful at sustaining itself through donations:

- The program must be aimed at end-users, in a substantive niche. That means having a potential userbase in the tens of millions, at least.
- It must be cross-platform (all the major operating systems).
- It must be *polished*. That means being easy to use for novices, especially for its primary use case. For instance, despite being so powerful, calibre is carefully designed to be very easy to use for the basic use case of add book, connect device, send to device, without needing to worry about any ebook related arcana.
- It must have comprehensive documentation/tooltips/demo videos/etc. to make it easy for novices to figure out things without relying on support. Because support is expensive.
- On a technical note, its code needs to be well designed to allow for easy maintenance and rapid feature development by a relatively small team. For instance, calibre has had weekly releases for over six years. There are very few software projects that can sustain this pace for years. And calibre does it with just one full time developer, me.
- The support infrastructure needs to be efficient. That means using lots of automation in the build system, the distribution servers, the bug report system, etc. For example, calibre releases are built and published for seven platforms automatically using a single command. Similarly, I can respond to most bug reports with just a couple of keystrokes.

All this is necessary to maintain an infrastructure lean enough to be supported by donations.

Sadly, the emergence of ebooks has, in addition to increased possibilities of literature and knowledge dissemination,

[1] Free, Open-Source Software.

also brought about measures of control and technical restrictions of reader/user freedom. To get hold of ebooks through purchase or library ”loans”, you often have to buy into a vendor ”ecosystem” that's locked up through digital restrictions management (DRM). What's your view on DRM and its role on the ebook market?

DRM is snake oil. It does not protect ebooks from piracy. Its only effect is to help create walled gardens and promote vendor lock-in. I strongly believe in DRM-free publishing, to promote which, my wife runs the Open Books website – a curated list of DRM free books.[2]

I have an e-ink reader that I love (a rooted Android device set up with Coolreader for epubs and APV for pdfs), but recently I have begun to read even complete novels on my smartphone. I still switch a lot between print and screen, though, and I admit to sometimes printing out digital articles to read on paper … What do your own reading habits look like?

I am a voracious reader, and have been one since childhood. For me, paper was not a realistic option, both because of its bulkiness and restricted availability. I was reading on PDAs long before e-ink based readers existed. Back then, I would read a mix of paper and digital books. Today, I read exclusively on my e-ink based device (I find backlit devices like tablets strain my eyes during extended reading sessions, so I stick to e-ink). I am a fan of History/Historical Fiction/SFF and I also read non-fiction books on widely varied subjects.

What are the future plans for calibre?

calibre development is very user driven, so I tend not to make plans for it. New features happen as they are needed by the user community. One large development that I personally intend to implement is allowing the calibre content server to actually edit your book collection (currently it is read-only) and also add the ability to read books directly in the browser through the content server without needing third party reading apps on your mobile device.

Sounds like nice new features! Since you mention this feature of direct reading through the browser, what do you think about the overall tendency in the networked society to rely on (personal) data stored in the cloud and web-based apps?

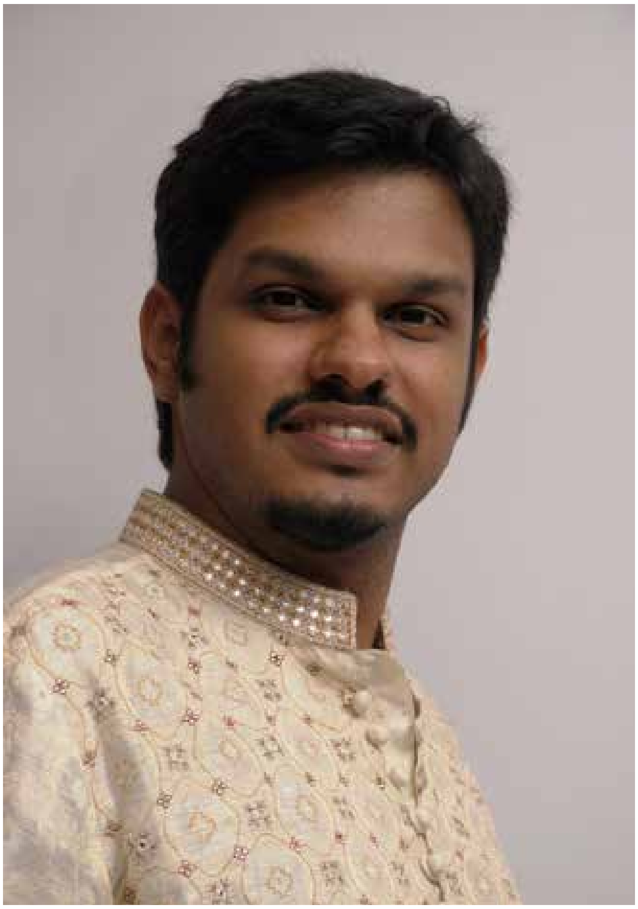
Don't do it :-) If you must have all your data available all the time, run your own server and store your data there. The calibre server is designed to allow you to run your own server, not store your data in somebody else's cloud.

Currently, most software companies are trying to create a future where they run all software and all you do is import and export your data, through more or less sophisticated interfaces, that they fully control. See for example, webmail, online office applications, cloud storage apps, etc.

In my opinion, as time passes, it will become easier and cheaper to own your own general purpose permanently network connected computer and run whatever software you like on it, at your discretion and fully in your control. See for example Sandstorm.[3]

So, running your own server (or one for your friends) can be considered an important act of empowerment in the increasingly networked, digital infrastructure. Hackers already do this to a large extent, but I guess the big challenge is to get users without programming/sysadmin skills to set up their own servers? Freedombox[4] also has a similar approach to that of Sandstorm, simplifying the user interface to the powerful, and free, communication protocols already out there.

Yes, the challenge is to make it easy for non-hackers to run their own servers. Hopefully, as software engineering matures that will happen.



Kovid Goyal.

[2] http://drmfree.calibre-ebook.com/.

[3] https://sandstorm.io/.

[4] http://freedomboxfoundation.org/.