

# Software Freedom Can Lead to Self-Reliance, says DeepRoot Linux Founder



Abhas Abhinav, founder of the Bengaluru based DeepRoot Linux, is an entrepreneur and hacker specialising in free software and hardware. For him, free and open source software (FOSS) is not just a professional choice, but a way of life that represents the very essence of personal freedom.

In an exclusive interview with OSFY, Abhas shares details about his initial encounter with FOSS and his subsequent free and open journey. Excerpts follow...

## Q What were your initial thoughts when you experienced the FOSS phenomenon?

I didn't realise it was a phenomenon at that time. I first learnt to use bash, Perl and vi on a UNIX system over very slow (2400 baud) dial-up connections. These were the entry points for me to a very rich world of possibilities. I was using Windows (3.1 and then '95 and NT4), but soon I realised that there wasn't much to learn, do or program on Windows. And so I began exploring other operating systems such as OS/2 Warp and finally, GNU/Linux.

In 1998, I started using GNU/Linux full time after installing it on my desktop. I think it was a Red Hat

## Q Can you recollect and share your first encounter with FOSS?

It must have been when I was still in school, back in 1994 or 1995 that I first began using some form of free software. I can remember two instances specifically.

I would use the console-based Lynx browser to access the Internet from a UNIX system in my father's office. Subsequently, I learnt to use vi and the UNIX command line too. In fact, I started liking Lynx so much that later on

(sometime in 1997/98) I programmed a DOS-based Web browser similar to Lynx using the Object Pascal programming language.

The second instance was when I learnt to use a tool called POV-Ray, which is a ray-tracer and can be used to design 3D graphics. The program came as part of a POV-Ray CD, which also had other tools such as modellers, viewers, animators and scene files that were designed and shared by other designers.

(4.2) release packaged on a CD that came with a book. This book was very comprehensive and it enabled me to learn Perl, awk, Emacs, Apache, Samba and LaTeX.

A feeling of great excitement gripped me as I had finally found an operating system with the boundless scope to explore, learn and build new things. I can't imagine what I would have done without this freedom to use, study and learn.

**Q Any individuals or platforms that played a significant role in strengthening your belief and connection with FOSS?**

Soon after I started using GNU/Linux, I discovered the GNU Manifesto. The politics of the software offered a completely different perspective on how to do computing. So, in that sense, reading essays by RMS (Richard Matthew Stallman) was the first 'non-technical' influence that I had. At the same time, I also read Eric Raymond's 'Cathedral and the Bazaar' essay and that had an influence on me as well.

I was a part of the LiveWire! BBS before we got Internet in '95. So, I learnt my email etiquette, concepts of online communication, collaboration and the spirit of healthy online discussions on various BBS, BharatNet and FidoNet forums. Most of the people who were on the BBSs at that time went on to be very active in the free software community too.

Also, I was a part of a community called the Microcomputer User's Club (MUC). It was an informal community of programmers and computer users who would meet over weekends, swap floppies of software, give technical talks and share knowledge with each other. Many of these people also had a deep interest in electronics and the DIY culture with respect to computers. And I believe, it is from them that I acquired the DIY attitude, which I later channelised into FOSS (and hardware work) once I got started with it.

After I started DeepRoot, there were several other people, events and

'platforms' that played an important role in reinforcing my personal belief in free software. This includes so many people in the free software community (students and team members who have worked with us and our community of users and customers) who have supported us by giving us their business and patronage.

“  
*A feeling of great excitement gripped me as I had finally found an operating system with the boundless scope to explore, learn and build new things.*  
”

**Q So has FOSS played a pivotal role in your professional life in any way? Do share the details.**

My professional life has only been about free software. While I was still a student, I used to manage servers for a friend, who later played a critical role in mentoring and helping me get started with DeepRoot. He ran a fairly large email and Web hosting company (around 1999/2000) and I also worked for some ISPs. All these servers were powered by GNU/Linux and free software. That is how I decided to work full time to start a company that would build products and provide such solutions.

All my 'professional' earnings have been out of FOSS. I began to understand the ideology of software freedom while running the business, and that helped me interpret and appreciate the philosophy and not just the technology. Understanding FOSS later enabled me to interact with all sorts of people professionally.

It has provided me the opportunities to do such a wide variety of things. It has enabled me to solve problems for a very large number of diverse organisations. And it has also helped me

to cultivate, and then hone my skills as a 'hacker'.

Today, if we have enabled others to learn about free software in so many ways, it is all because we have had the opportunity to learn about those things from others. Otherwise, it would have been impossible to build the sort of solutions we have built over the past 18 years.

**Q Has FOSS also affected any aspects of your personal life?**

FOSS has had a great impact on me personally as well. For me, free software is not just a professional thing to do... it's a way of life. The free software movement has taught me important lessons on reciprocity, fairness, sharing-alike, on respecting others, on cultivating empathy and, most importantly, on the value and essence of personal freedom.

I have learnt the value of having a strong (and recognisable) value system and standing by it — especially in the face of resistance, inconvenience and discouragement.

Adopting free software as a way of life has also made it possible for me to understand and appreciate other social movements and learn from them as well.

Free software has enabled me to build a personal ethic of sharing knowledge. Throughout school and then college, we are not really taught to share knowledge — not formally at least.

**Q How do you view the current status of FOSS? Has it evolved as per your expectations?**

The current status of FOSS is much below my expectations. I feel this is primarily because of two factors: (a) we routinely choose convenience over freedom or control in technology, and (b) we are not always mindful about the choices we are making, and whether they are consistent with whatever we want to be optimising for in our lives.

Today, we don't often see the dangers of proprietary software and

how it threatens the way we compute. Our perspectives are clouded by the ‘coolness’ or ‘utilitarian’ value of our choices. Or how much ‘convenience’ they provide in some area of our lives. At least ten years back, the situation was not so bad because we didn’t have the cloud, we didn’t have smart gadgets and we didn’t have various smartphone application repositories. Free software presented us with the only paradigm whereby we could take complete control and ownership of the technology that affected our lives and work.

The cloud enables us to use proprietary software that runs on someone else’s computer. Smart gadgets are nothing but hardware + software combinations with the software providing the ability to track us. Smartphone applications are another way proprietary software creeps into the daily life of even the most ardent free software user. Finally, there is proprietary JavaScript that runs in our browsers which is a silent threat to our computing freedom.

The reason I say that the current status of FOSS is below (my) expectations is because when you observe the amount of proprietary software that we use today, it is actually far more than what we used even ten or 15 years back. And for some reason that doesn’t bother us as much.

While FOSS has enabled the creation of so much wealth and software, and while it is impossible today to do anything that is not derived from or based on FOSS, we still use much less of it.

We might not be able to remove proprietary software from our lives completely. But by being mindful, by building and then following an identifiable free software value system, and by reminding ourselves of what really matters to us, we can limit this usage (and its impact) to the minimum.

### Q How do you see FOSS evolving from here?

Free software is going to be even more

important in the future. As we race to protect our privacy, and to safeguard our freedom to study, learn, extend, fix, modify, share, embed and host software, we will discover that we can’t do all of these things without also caring about free software.

In the future, we might have to focus a lot more on doing our own computing and building our own technology. As self-hosting becomes the norm and as DIY becomes a ‘default choice’ for some people, free software should become more relevant, important and indispensable.

“  
*While FOSS has enabled the creation of so much wealth and software, and while it is impossible today to do anything that is not derived from or based on FOSS, we still use much less of it.*  
 ”

Unless we carefully choose the hardware we purchase, we will slowly run out of places to run our software on. So many products today are combinations of hardware and software. However, very few of them provide us the freedom we deserve. The future of free software, in general, is also dependent upon our ability to run free software everywhere we want to. And to do that, we need to either choose our hardware carefully or design hardware in a manner that it can provide this very important freedom to its users.

### Q What is the role of India (and Indians) with respect to FOSS?

Since India (and Indians) are the biggest consumers of software across all devices and computing systems,

we can choose to have the most (or least) impactful role with respect to FOSS.

There are a few aspects where FOSS is even more relevant to India.

If we want to create a level playing field for everyone to build software and hardware and if we want to see some sort of ‘equality’, then there is nothing better than free software to achieve this.

Using and promoting free software is a way to create equal opportunities for everyone - irrespective of their technical, social or economic background.

If we can build, service, extend, share and maintain our own software and hardware, software freedom can lead to self-reliance. Software freedom provides us with an opportunity to learn and build skills and competency. It is up to us to use this opportunity to create a culture of self-reliance. If we don’t do that (i.e., if we are just consumers of FOSS), then we are not even drawing the benefits from the (technological) freedom that we have.

There are problems, constraints and opportunities that are uniquely Indian in nature. If we choose to solve these problems by using and building free software + *libre* hardware, then that can help us to really own the solution that we build. That solution will not be the domain or property of a single person or company – it will be the users and community that can really own it and draw benefits from it.

We can’t expect someone from outside India to understand or appreciate these problems enough to localise the solutions for them. And in that respect, there is no better way than to use and build free software.

In a nutshell, it is up to all of us to make the best use of the opportunity that free software provides us to solve our problems ethically and elegantly. That is how we can play a significant role in the global free software ecosystem as well.

## Q Do you see India gaining more importance in the global FOSS ecosystem?

Yes, of course. There are two aspects to this.

There are many alarms going off right now that, gradually, lead to the realisation that trusting proprietary software (and network) providers with our data is not a good thing to do. Some of these problems are global in nature, while some of them don't affect us yet.

Indians can gain more importance in the global FOSS ecosystem by pre-empting these challenges in our own society and solving the major problems that everyone is facing. A simple example is that if we can use and promote federated services on a large scale, it can offer a sustainable solution to this problem of websites tracking us and imposing conditions on access to our own data.

There are many other societies and countries that look up to us on how to solve these problems in an effective manner. Hence, by first using free software to solve these problems at our end, we can also provide leadership to everyone else and lead by example.

Finally, I would like to add that we need not just do things that are important for the global free software ecosystem. That could be a long-term side effect of the things that we do anyway. It need not be a specific goal.

## Q Your favourite authors or books related to FOSS?

This is a very difficult question to answer since much of what one can learn about free software is not derived from books. However, I would like to mention three resources that can provide good starting points to others.

'Hackers: Heroes of the Computer Revolution' by Stephen Levy is an amazing book that details the history of the computer revolution and illustrates the 'hacker ethic' in the best possible way.

The source code, documentation and systems written and designed by D. J. Bernstein (DJB) have been inspiring.

While I don't use as much of DJB's software as I once used to, it has had a huge impact on my understanding of how software and systems should be designed and documented. In a way, his code is still poetry.

'Coders at Work: Reflections on the Craft of Programming' by Peter Seibel is a book about the thought processes and inner workings of some of the seminal hackers of computing.

“  
*Using and promoting free software is a way to create equal opportunities for everyone - irrespective of their technical, social or economic background.*  
 ”

## Q If you were a youngster today (fresh out of college), what would you do differently with respect to FOSS?

I would focus more on writing documentation. I have learnt the hard way that to develop and service free software with excellence, documentation is extremely necessary—all the more if you're going to dedicate your life to working on free software. It is well known that doing things is more exciting for us than documenting them. But without documentation, there is no way to illustrate and explain what we are doing and to help others do it as well.

I would generalise and share a whole lot more of whatever I learn in the course of work. It is very unfortunate that though we (at DeepRoot) have built some extraordinary solutions over the past 18 years, we have not been able to make them generic or share them.

I would focus as much on building products as creating markets and sales

channels for them. This is important for any business, but it is especially important for free software businesses.

I would not waste any time pursuing venture capital — it is highly unlikely that you will find someone who will look beyond monetisation to truly understand the long term value and impact of free software, and invest in it. And hence, any time spent pursuing this option (at least in India) is a waste of valuable time.

I would devote much more time to participating in the local community and in user group activities. It was an error of judgement to prioritise 'work' at the cost of volunteering for the community.

## Q Earlier, many firms and luminaries dismissed FOSS as a fad. Now, almost all greet it with open arms. What's your take on this?

Social movements are not fads. If someone has dismissed the free software movement as a fad, then they have only viewed it from a short-term or technological perspective. They have not yet been able to appreciate and understand the relevance and ideology of the free software movement.

At the same time, sometimes it takes time for the mainstream to catch on. Unless someone has owned and bought the problem, they can't appreciate the solution. So promoting free software as a solution to certain problems will not be received well unless someone has experienced or internalised the problem first-hand.

I think that's fine. It is, then, the responsibility of those who *do* understand to talk about it and create greater awareness and understanding about it.

## Q What would be your advice to youngsters who may still be thinking about whether to build their careers in FOSS-related tech or not?

If they want to optimise for learning and experience the thrill of building things and solving problems, then I would definitely advise them to consider free



software as a way of life. I don't see any other reason or way of building things and so if someone asks me, I will always say that they should invest in using and building free software as a career.

I would advise young people in school, college and beyond, to focus more on the sort of things they can do and the depth at which they can learn instead of finding quick or easy ways of doing something. By just consuming what is packaged by others, we also eliminate an opportunity to do things ourselves and hence, learn. If our ability to create something and solve problems is to be based on what we are capable of doing on our own, then choosing free software and hardware to build these solutions is the best way to learn.

Very long ago, I would ask students working with us to follow the 'Linux from Scratch How-to' to manually build their GNU/Linux system without the aid of a distribution. So while it took them a few days to build a usable desktop (as compared to a few minutes that it would take to install Debian on it), the insight and learning they got was unprecedented.

Additionally, I see a lot of young people using free software on proprietary operating systems such as Windows or MacOS. While that might still get the job done, I don't understand how that is the most effective way of building or using free software. There is no limit to what one can do on a GNU/Linux computer. And I would encourage everyone to use GNU/Linux. There is help around if you get stuck – just don't give up when you face a challenge.

Lastly, I would encourage young people to get more involved with hardware experimentation. By understanding how hardware systems work and the role software plays in their functioning, we can build more hardware capable of running free software and can protect our freedom

to compute. Each 'smart device' that we see around us is often easy to build if we choose to do so. And there is still no better way to do that than by using free software and hardware.

**Q Any advice for academicians?**  
To academicians who teach or deal with computing related subjects, I would suggest the following.

**Create a culture of learning by doing:** To be more effective at teaching a practical subject, one needs to have experience and skills. To gather experience and cultivate these skills, one needs to do things. One cannot do things if one is just a consumer of technology. A very good way to do tasks is to build things yourself.



*If someone has dismissed the free software movement as a fad, then they have only viewed it from a short-term or technological perspective.*



By building things yourself, you can acquire valuable skills, experience and insights. And these can, in turn, help you become more effective at teaching.

But, how can you build things with 'black boxes'? You can't! To learn to build things, you need tools, directions, knowledge and opportunity. Free software is an ideology that can provide all of these to you. With free software, you have unrestricted access to the resources.

**Choose self-hosting:** This means that if we need some value from a software or service, instead of deriving that value from proprietary software hosted on the cloud (or only available as a service), we instead, deploy it ourselves on computers that are within our control.

Today, there are very few cases for which you need to use proprietary software (on-premise or on the cloud) in an academic setting. Everything that you need to do can be done very well using GNU/Linux and free software.

**Choose free software:** When we evaluate our choices of which software, operating system or tool to use for self-hosting the service or solution, we should be focused on choosing free software.

If we chose to self-host using proprietary software, we would have the following constraints:

- We would have no insight into how things work internally (i.e., a wasted opportunity to study and learn because there is no source code provided).
- We would have constraints on how we can use software, for what purpose, with whom, where and in what quantity.
- We would not be able to modify the software to extend it, learn from it or fix it (because there is no source code provided).

So much free software that we use today has originated in academic institutions. Once we start using it, the next step of contribution is more natural and obvious.

**Build it yourself:** The next step is to do the self-hosting ourselves. Outsourcing the process of self-hosting (even if it is with free software) will not yield the results we want.

**Inspire and lead students by example:** Lastly, inspire students to learn by putting in similar efforts. By leading by example, we set the benchmark very high. By teaching out of experience, we can help them approach more challenging problems (than what they pursue now).

**Q Any advice for the media?**  
Don't discount and ignore people, projects and businesses because they are 'too idealistic'.

Don't use the same measure of 'success' for everyone. For some, success is measured in terms of their financial impact; for others, it is measured in terms of the value of their creation; and for others, it can also be measured by what means and values they adopt to move towards their goals (and how or whether they remain true to them). While it is good to cover 'successful' outcomes, it is also important to cover moderate outcomes or failures because they offer greater lessons and opportunities for us than the successes.

Philosophical and ideological discourse in technology is also important. We should raise and highlight issues that address moral dilemmas that people face. We should not cover technology just from the utilitarian perspective—as a means to an end.

Many times, the content I read in mainstream media reflects a complete lack of awareness about concepts such as free software, creative commons, sharing knowledge, ethical technical practices, respecting users and their rights/freedom, and so on. Such reporting just reflects the opinions and positions of various proprietary

software companies, associations and lobbies.

I feel that if more media persons were aware of the free software movement and could reflect on the technical and ideological positions we take in the community, it would help them assess and report on issues in a more balanced and inclusive manner. Sometimes, the lack of this understanding and awareness reinforces popular misconceptions about the technology.

“  
*Each 'smart device' that we see around us is often easy to build if we choose to do so. And there is still no better way to do that than by using free software and hardware.*  
 ”

**Q Any advice for entrepreneurs who are doing business around FOSS, or would want to?**

If you want to eliminate proprietary software and devices that don't respect your freedom and privacy, you have

to also build such software and hardware. It can't be somebody else's problem. The other person's need for freedom, respect and privacy is as great as ours. Proactively building products and solutions that do this by default is a way to “be the change that we want to see.”

We can get rid of the problem of 'black boxes in our lives' by using free software and self-hosting. Businesses are at the solutions end of these problems.

The challenges of building a business remain the same, irrespective of whether the primary focus of your business is free software or not. You still have to build and lead a team, you still need to find customers to sell to, and you still need to reach out so that others can discover your offerings. However, choosing free software can greatly ease these challenges.

A free software business charges people for the value it provides. This value is not derived from restricting access to the software or its source code – by definition, a customer has the freedom to use, study, modify and share the software. Hence, the customer is not paying for access to the software.

There are, however, things that customers will definitely pay for: time-to-productivity (how soon can I start using the solution), support and services (who can I call when I have a problem), migration (can you help me shift my work from X to Y), documentation and training (can you also show me/teach me how to do things), hosting and maintenance (can you help me deploy and manage the software as well), and customisation (can you get this software to do that). These are some of the things a free software business can get creative about. **END** 🐧

As suggested by Abhas, here are some resources that can provide valuable insights into FOSS:

1. 'Hackers: Heroes of the Computer Revolution' by Stephen Levy; [https://en.wikipedia.org/wiki/Hackers:\\_Heroes\\_of\\_the\\_Computer\\_Revolution](https://en.wikipedia.org/wiki/Hackers:_Heroes_of_the_Computer_Revolution)
2. Source code, documentation and systems written and designed by D.J. Bernstein; <https://cr.yp.to>
3. 'Coders at Work: Reflections on the Craft of Programming' by Peter Seibel; [https://en.wikipedia.org/wiki/Coders\\_at\\_work](https://en.wikipedia.org/wiki/Coders_at_work)

Abhas provides some pointers on why freedom matters even for students and what it means to be a hacker at <https://abhas.io/students-hackers/>.

For those who have teaching roles in academic institutions (especially faculty members in engineering colleges), Abhas offers a roadmap on why it is important to choose free software to optimise freedom and learning. The presentation is available at <https://abhas.io/for-teachers/>.

For entrepreneurs who are doing business around FOSS, this resource may help: <https://freesoftware.business/>.

In his presentation at <https://abhas.io/no-black-boxes/>, Abhas explains the problem of 'black boxes in our lives' and how we can get rid of them using free software and self-hosting.

Abhas can be reached at [abhas@deeprooot.in](mailto:abhas@deeprooot.in). You can know more about his work at [abhas.io](https://abhas.io).