

开源软件设计与开发

52195100001 佟兴

开源，全称为开放源代码。目前，市场上开源软件层出不穷，很多人可能认为开源软件最明显的特点是 free，但实际上并不是这样的，开源软件最大的特点应该是 open，也就是任何人都可以得到软件的源代码，加以修改学习，甚至重新发放，当然是在版权限制范围之内。

开源软件其实面向的用户有两个群体，一是程序员，他们最关心源代码，能不能进行二次开发利用；二是普通终端用户，他们只关心你的软件功能够不够强。

开源软件的重点应该是在“开放”，是接纳、包容和发展，求同存异，互利共赢，才是开源的本质。

有人问，开放软件的源代码，肯定就会有技术高手利用你的源代码开发新的产品。是的，开源就是要用户利用源代码在其基础上修改和学习的，但开源软件同样也有版权，同样也受到法律保护。对开源软件来说，如果今后发现滥用开源产品或在开源产品基础上加以发展任何派生版本、修改版本或第三方版本用于重新分发，都将会受到法律的制裁，支付侵权赔偿金。用户在使用开源产品时，不但需表明产品来自开源软件和注明源代码编写者姓名，而且还应把所修改产品返回给开源软件，否则所修改产品就可视为侵权。现在，国内的盗版很泛滥，即便闭源的软件，都会被肆意盗版甚至篡改版权，开源软件就更别说了，篡改版权就是个查找替换的简单操作而已。版权意识的淡漠才是国内开源发展的最大障碍。

开源软件在国内起步晚，但发展很快，将来肯定是业内主流。那些打着开源的幌子貌似开源，将其核心代码加密的软件必将引起公愤。开源真实含义应是利用其源代码加以修改和学习，在理解开源真正含义后，那些造假、侵权、违法的行为和事件必定减少。

由软件精神包括两个方面

1. 用户的自由：软件不应该干涉和限制用户对机器的完全控制。在 fsf 宣称的自由软件哲学里面这被叫做四个自由。当然，这包括用户任意修改软件行为的自由和再发布的自由，因此在这个层面上自由软件必须是开源的，但不必须是 copyleft 的。

2. 知识传播的自由：copyleft。一个思想只有传播才能体现价值，只有整个社区的合作才能带来社区的发展。copyleft 精神虽然是一种理想主义，但是这种理想主义是具有现实意义的，尤其是，在著作权和专利成了阻碍创新的壁垒，而不是鼓励创新的动力的今天。

虽然逻辑上自由软件和商业化不冲突，但实际上自由软件精神是和商业化相悖的。

1. 对于用户的自由：虽然用户愿意为自由软件付费，但是商业软件不会允许用户拥有再分发的自由。而且，为了进一步保障自己的利润，商业软件通常会在很大程度上限制用户对软件行为的修改。

2. 对于 copyleft：商业软件当然不会愿意进行 copyleft 授权，而且使用 copyleft 授权的软件具有“外部性”。

我所理解的自由更多的是对知识自由传播的自由，也就是 copyleft 的自由。因为现在的 copyright 模式催生了大量的专利流氓和畸形的出版商，尤其是学术出版领域。而知识传播的自由自然会带来用户的自由。

前一段时间有个新闻，说某大学因为知网授权费太贵而停止付费，可见学术出版商的收费是有多么惊人，而学术出版之所以能够收取如此多的授权费用，是因为著作权在出版商手中，而不在作者手中。甚至作者为了看自己的文章，还需要向出版商付费，简直是滑稽。

同样的事情在软件行业也大量存在，程序员亲手写出的代码，著作权居然不在程序员本人手中，程序员不能随意使用雇主的代码，即使那是他自己写的。再比如，软件领域的专利流氓，实际上创业者完全无法绕过这些专利，只能向专利持有者付费。这并不是应该是正常的商业模式。

开源是“免费”的，而“免费”往往是最贵的，免费同样不是目的，其背后创造的价值才是。无论是 **Free Software** 还是 **Open Source**，其发展也从未脱离开与商业的关系。

在开源运动的早期，**Microsoft**、**Oracle**、**IBM** 都曾经是开源软件的公开反对者，甚至不乏批判的声音。但时间的推移让我们看到了开源模式最终获得了胜利，过去一年中，整个软件行业最大的新闻之一应该就是 **Microsoft** 收购了 **GitHub**，前者曾经是开源联盟的头号公敌，而如今，却与 **Google**, **Facebook**, **Apple**, **Amazon** 等等所有科技巨头公司一起全力开源各个项目。而基于 **GitHub Octoverse 2018** 的数据，**GitHub** 上最热门的 10 个项目中只有 2 个项目是由社区驱动的，而其余前 8 个项目皆是隶属于大型科技企业。

开源绝非仅仅是慈善，开源软件本身的免费传播，却孕育出开源软件之上的服务（如软件服务、书籍、教程、咨询、培训、展会等）都存在商业价值，早期如 **VA Linux**、**Red Hat** 的公司也在 1999 年相继上市，当下也有 **MongoDB** 等企业依然在快速发展。开源软件公司的收入帮助开发者可以继续维护、发展、运营整个社区，并面向企业提供高级定制化软件服务或 **SaaS** 服务获取收入。

参与项目：**TiDB** 的向量化计算

TiDB 的向量化计算是在经典 **Volcano** 模型上的进行改进，尽可能利用 **CPU Cache**, **SIMD Instructions**, **Pipeline**, **Branch Predication** 等硬件特性提升计算性能，同时降低执行框架的迭代开销。