

1 开源理解

开源是互联网时代的一个福利,因为开源,让需要的人可以查看、复制、学习、修改或者分享代码,降低了软件行业的准入门槛,以及软件开发的壁垒,使得更多的人可以拥抱开源,参与到开源中来,而 Git 的出现以及 Github 社区,使得全世界的人们通过互联网共同协作开发软件,这是一件很有意义的事情,在很大程度上促进了软件的发展、让计算机编程更加的平民化、大众化。

Git 工具是一个十分神奇的东西,对比使用过 SVN 协同开发软件,同样的项目在 SVN 下总是出现各种问题,当版本管理软件转向使用 Git 后,大部分问题都得到了解决。同时也很好的加强了团队的沟通效率。

本学期参与的开源项目为 TiDB(<https://github.com/pingcap/tidb>),通过参与这个项目,第一次接触并且学习使用了 Git,从最开始的 fork 仓库到后面的 commit,以及完成需要开发的任务后提交的 pull request,让我对整个团队协作开发的流程有着更加清楚的学习和认识。

随着互联网、云计算、大数据、人工智能、区块链等前沿技术的陆续诞生,开源软件也在全球蓬勃发展,开源社区使全球信息技术领域发生了全局性的、持续的重大变化,并在社会基础设施建设方面发挥着愈加重要的作用。

开源社区一般由拥有共同兴趣爱好的人所组成,根据相应的开源软件许可证协议公布软件源代码,由于开放源码主要被散布在全世界的编程者所开发,开源社区就成了他们沟通交流的必要途径。以 Github 社区为例,大批周边扩展服务被建立起来,构成了一个极具活力的生态圈,开发技术人员不仅可在 Github 上参与开源项目,更可建立社交圈子,促成开放的分布式协作模式。

开源社区的最主要特征是:团队协作、个体平等、主动贡献,这也是开源精神的主要内涵。开源社区具备很强的利他主义 (Altruism) 精神,参与到开源软件开发并把源代码开放给大家共享的开源社区成员,一般称之为贡献者 (Contributor)。贡献者通过参与开源社区的开发,一方面得到了锻炼成长的机会,也有助于解决自身工作中遇到的技术问题,另一方面开源项目也有机会产生较大的商业价值,而商业化进程中就又需要这些贡献者担任顾问以推进技术落地。

开源的本质是共享技术,而“技术”是生产资料的组成部分 (拥有技术的科学家、工程师们可以看成是技术的载体),因此开源的社会意义实际上是实现了生产资料的共享。开源的商业意义则在于,通

过共享技术、推广技术标准从而打造了一个多方利益共同体，利益相关者的数量对商业生态系统的形成至关重要，体系内部的分工环节越多越细，说明系统内部的利益主体就越多，从而这套生态系统的市场占有率可能就越大，在与其他闭源的技术竞争的时候就越有力量。

从安全的角度，开源技术也丝毫不逊于闭源技术，甚至更有优势。开源软件的代码可以被任何人阅读，也有可能被黑客关注，分析可能存在的漏洞并进行利用，但在软件行业的经验教训中，开源软件的漏洞远远少于闭源软件，因开源软件的代码可以被全行业审计，被不同的开发者修订和安全加固，在安全方面的风险会大大降低。举例而言，闭源的 Windows 系统与开源的 Linux 系统相比，公认都是 Linux 系统更胜一筹，正是因为 Linux 系统的源代码公开、有源源不断的开发者加入对其优化，才使它的漏洞尽早被发现，避免遭受攻击。此外，开源软件本身一般不包括业务逻辑，不会影响到业务合规、资金安全和用户数据隐私等问题。

任何协议或法规的实现，都会秉持某些精神或理念，就连开源这项行为也不会例外。正如法规条文的制定要符合法律精神这一大前提一样，开源协议的制定与执行，当然也应该秉持并符合一种精神—开源精神。

什么是开源精神？开源精神是造福大众的开源精神，而不是大众服务你，大众造福你的开源精神。

什么是开源精神？开源精神是知识共享，互惠互利的开源精神，而不是知识垄断，弱者恒弱，强者恒强的开源精神。

什么是开源精神？开源精神是扶危济困，损有余而补不足，先进带动后进的开源精神，而不是劫贫济富，以不足而奉有余，先进压制后进的开源精神。

什么是开源精神？开源精神是公正公开，维护开发者与使用者利益的开源精神，而不是闭塞停滞，众乐乐不如独乐乐，不管他人死活的开源精神。

开源精神，应该是自由，创新，团结，互助，友爱，积极，进取的开源精神，而不是垄断，封闭，孤立，排异，冷漠，畏缩，退步的开源精神。

无论何种开源协议的创立或使用，都应该是为开源精神保驾护航，而不能反过来阻碍甚至摧残开源精神，折磨乃至摧残开源者。

正如同西方法系中，既会有法官根本客观法律条文来确定具体法律规范的实施，也会有陪审团制度来确保法律审判是民主的、人性的法律精神体现一样。开源协议，就应该为开源精神服务，哪怕在某种情况下协议会与精神存在冲突，也应该协议是遵循开源精神，确保真正开源者与开源项目的利益，而不是为了几 KB 的协议文件，至开源者与开源项目于死地而后快。

譬如猎枪，本是猎人对付豺狼的工具，应改为消灭豺狼服务。但如果他落入了豺狼的手中，豺狼反举起它消灭起猎人来。你却还为猎枪的使用而欢呼雀跃，高叫什么打的好啊！打的好！杀的妙啊，杀的妙！。那么，你不是先天智障，便是某些幻化成人形的狼精豹怪罢了。

开源协议，就应该也必须为开源者服务，为开源精神服务，如果反过来，开源精神为开源协议服务，那就本末倒置，成了为形式而内容，失去了协议的存在价值和意义。

宙斯立有天条，人间不能获得火的使用，这是法律（协议），普罗米修斯为了人类，毅然破坏协议

盗天火下凡间，所以民间奉他为英雄。诺丁汉郡长立有法规，伍德森林中盗贼违法，这是法律（协议），但罗宾汉为救理查王多次与他对抗，并最终将他打败，所以民间奉他为英雄。

无论西方东方，一旦遇到协议与事实发生冲突，事急从权，这都是合情合理的必然选择。至于这些行为，能不能得到大众的承认，就看你是损有余以补不足的为民请命，还是损不足以奉有余的为官请命了。

如果真是为了保护开源软件的正当权力，就算在某些手段上与开源协议存在冲突，那又有什么呢？在法律无法保护你的情况下，你就应该由然自若，坚定不屈的将有利于开源项目的行为进行到底。因为你们问心无愧，因为现状不允许你们充大头。一旦条件与环境允许时，你再按照规范办，按照协议办，又有什么不可以呢？

所以，笔者一直强调在某些国情特殊的地区，开源协议不单要和软件专利相结合，而且更要竭尽一切手段的保护自己，避免自己成为业界巨鳄嘴下的牺牲品，就算闭源部分核心代码也在所不惜。全因为在这些地方，你的开源项目所能依靠的，除了你自己以外，就再也没有其它什么了。不光为了你，也为了这项目本身，因为在某些时候，某些利欲熏心的家伙，即便赤裸裸的剽窃了你的代码，也敢说抄了他们，而将你送上法庭，至于万劫不复之境。前一阵笔者曾建议新近出现的国人 Java 项目 Douyu 作者，如果有闲钱时可以申请个技术新型专利，为的也是这么一事情。

但是，如果你反其道而行，如果人云亦云，脑子一热，听到有人高喊某某不是真心帮人，家里还留下二百块钱生活费，便将家底一股脑丢了出去，害得自己冻死街头，落得此地从此再无一个好人，害得本来依赖你帮助的人也举步维艰，你说你是可怜，可叹，还是可笑，可悲？

任何死板的遵循开源协议，却失去开源精神的行为，都是疯狂的，片面的，无知的，就如同肉体没了灵魂，就如同法院没了陪审团，就如同李云龙再不敢亮剑。

在目前的特殊环境下，指望开源协议起到保护开发者，保护开源项目的作用，无异于痴人说梦。与其坐以待毙，倒不如事急从权，将某些核心代码彻底封闭，一方面发展自己，一方面等待能够全部开放的时机。

有鉴于此，为了适应国情，有了中国开源者的未来。发展出我们中国自己的，有中国特色的开源协议，再不是一个笑话。就在此时，就在此刻，应该被认认真真，端端正正的，摆到国内本就少得可怜的开源者们的议事日程上来了。

如果开源协议在具体环境下适用，就该遵从协议，如果协议在具体环境下不适用，就该依照开源精神扬弃协议，事实就是这么简单，也只有这样中国的开源者才会有生存的可能性。

否则，遵守开源协议的人都被豺狼咬死了，协议就彻底成了擦屁股纸，连存在都不会存在，又能指望它干些什么？！这也是自然法学派，提出恶法非法的根本道理。

如果说开源精神是虎，是我们所要维系与达成的目标，那么开源协议就是猫，所谓照猫画虎，开源协议应该是让我们距离体现开源项目本质的，也就是开源精神越来越近的工。而不能背道而驰，让我们画虎不成反类其犬，让开源协议成为开源项目的绊马索，让开源协议成为豺狼指责我们，消灭我们的猎枪，最后反将开源项目带入灭亡的深渊中。

岂能尽如人意，但求无愧我心，只要开源者清楚自己的所作所为究竟为了什么，我们的付出究竟为了谁，他人的风言风语，又算得了什么？—更何况，你所将面对的，或许还是群你和他讲道理，他和你耍流氓，你和他耍流氓，他和你讲的道理的无耻之徒。

开源，不仅是种行为，更是一种信仰。

2 开源贡献

- 参与开源项目 TiDB: <https://github.com/pingcap/tidb>
- 开源项目 Pull request: <https://github.com/pingcap/tidb/pull/13170>
- 自己的开源项目: <https://github.com/CCChieh/ginHelper>

本学期参与并贡献的开源项目是 TiDB (<https://github.com/pingcap/tidb>)，我提交的 Pull request 是“expression: implement vectorized evaluation for 'builtinCastJSONAsTimeSig'” (<https://github.com/pingcap/tidb/pull/13170>) 即对一个表达式运算进行向量化。

由于有许多不懂得地方，所以前前后后总共提交了 22 次 commits，每一次提交都会有社区的 Reviewer 过来 review 我的代码，对我的 PR 的代码提出修改的建议。在第一次 PR 的时候，没注意提交了别人已经实现的模块，Reviewer 很细心的指出我实现的功能已经实现。当最后一次得到 Reviewer 的 LGTM 后我是很开心的，这也意味着我的代码成功的合并到了 TiDB 的 master 中，成功的完成了一次开源合作，这是我第一次完成 PR，在这之中也学习到很多。

在这之后我自己也开始了几个开源项目的开发，第一个完成的开源项目是 ginHelper (<https://github.com/CCChieh/ginHelper>)，ginHelper 重定义了一种代码低耦合、低冗余并且高效率的 Gin 开发方式。

开发 ginHelper 的动机又得从另一个开源项目说起，这个项目就是 goSpree (<https://github.com/CCChieh/gospree>) 这是我目前正在开发的一个项目，主要是想要用 go 开发一个自己的 blog，用来记录自己的学习笔记或者心得等，当然 goSpree 目前的完成度很低，打算在寒假期间完成基本功能的开发，在开发 goSpree 期间主要使用了 gin 作为 web 开发的框架，但是每次写 gin 的 handler 还有 service 的时候总是有很多的代码重复，所以便实现了一个 ginHelper 来解决这一问题。

目前 ginHelper 支持的功能有：

- 利用反射自动检索某一包内所有的 HandlerFunc(即 gin 的 Handler) 添加到 Gin 的路由中。
- 引入 parameter 接口的概念，通过定义实现该接口的结构体后可以自动生成一个 gin.HandlerFunc，利用该特点可以极大地降低 handler 和 service 的代码重复。

最终想要实现的是一个类似 swagger 的自动生成的 api 调试工具

3 课程反馈

个人认为，本课程 git 的教学可以放到课堂初期，然后初期刚接触 git 的时候可以使用一个简单的教学例子来完成 git 的演示，让同学都参与到这个 git 的入门教学中。之后再引入那些项目让同学进行实战的话效果会好很多。