

《开源软件设计与开发》课程总结

姓名：何家源 学号：51194507004

1、开源理解

我对开源精神的一些理解

“开源”全称“开放源代码”(open-source)被定义为描述其源码可以被公众使用的软件，并且此软件的使用，修改和分发也不受许可证的限制（定义来源于百度百科）。我认为开源不仅仅是一种技术手段，更是一种精神。开源与知识产权的利用不会让任何一个搞技术的人独善其身。目前全球科技性公司，特别是大型公司，仍然是处于垄断地位，但垄断的方式转变为利用“知识产权和专利”进行霸权行为。

垄断不违法，但通过垄断进行不正当的竞争即违法。这是某些国家法律上的称述，我很赞同。“专利和知识产权”，作为资产是需要保护的，即便形成垄断局面。但通过这种局面，剥夺甚至禁止其他人进入竞争的权力，则是我们需要抵抗的。有人说这和开源有什么关系，不妨思考一下微软的 Windows 和 Linux 在办公和生产系统中谁更好使？不可否认在家庭娱乐方面微软更占有。但事实是，中国绝大多数社会生产生活中，使用的是 Windows 平台，而不是 Linux。有些人说微软的系统是封闭的，Linux 是开源的，所以 Window 更安全。没有开源的精神，没有开源的认识观，则会导致这类迷信式的错误，而且会被垄断企业所利用。开源并不等同与不安全，相反闭源的 Windows 系统，用户并不知道系统内部的细节，那微软是否在通过系统收集用户的信息呢？相信除了微软公司的内部人员并没有人清楚。

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”的精神。开源精神是知识共享，互惠互利的精神，而不是“知识垄断，弱者恒弱，强者恒强”的精神。什么是开源精神？开源精神是扶危济困，损有余而补不足，先进带动后进的精神，而不是“劫贫济富，以不足而奉有余，先进压制后进”的精神。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精神，而不是“闭塞停滞，众乐乐不如独乐乐，不管他人死活”的精神。开源精神，应该是自由，创新，团结，互助，友爱，积极，进取的精神，而不是垄断，封闭，孤立，排异，冷漠，畏缩，退步的精神。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。

2、开源贡献

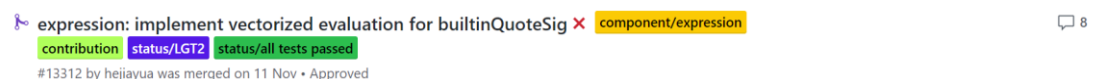
在本学期的开源课程中，我参与了 Tidb 开源项目，并在项目中提交了一个 pr，由于本人没有经验因此选取了一个 Tidb 中一个需要向量化的函数“builtinQuoteSig”。Pr 的具体链接为：<https://github.com/pingcap/tidb/pull/13312>

本人向量化的代码如下：


```
func (b *builtinQuoteSig) vecEvalString(input *chunk.Chunk, result *chunk.Column) error {
    n := input.NumRows()
    buf, err := b.bufAllocator.get(types.ETString, n)
    if err != nil {
        return err
    }
    defer b.bufAllocator.put(buf)
    if err := b.args[0].VecEvalString(b.ctx, input, buf); err != nil {
        return err
    }

    result.ReserveString(n)
    for i := 0; i < n; i++ {
        if buf.IsNull(i) {
            result.AppendString("NULL")
            continue
        }
        str := buf.GetString(i)
        result.AppendString(Quote(str))
    }
    return nil
}
```

被合并的 Pr 如下：



下面是跟 Tidb 项目中的 reviewer 交流的过程：

 **zz-jason** reviewed on 9 Nov [View changes](#)

expression/builtin_time_vec.go


790790}

791791


792792func (b *builtinNowWithoutArgSig) vecEvalTime(input *chunk.Chunk, result *chunk.Chunk) error {


793793return errors.Errorf("not implemented")

793793nowTs, err := getStmtTimestamp(b.ctx)

 **zz-jason** on 9 Nov Member + 😊 ...

@zz irrelevant change?

 Reply...

 **XuHuaiyu** reviewed on 7 Nov [View changes](#)


expression/builtin_string_vec.go Outdated

478+ result.ReserveString(n)


479+ for i := 0; i < n; i++ {

480+ if buf.IsNull(i) {


481+ result.AppendNull()

 **XuHuaiyu** on 7 Nov Contributor + 😊 ...


This is different from the non-vectorized implementation.

 **hejiayua** on 7 Nov Author Contributor + 😊 ...

@h I resubmitted after I changed it, please check it out.

 **qw4990** commented on 11 Nov Contributor + 😊 ...

/rebuild

 **qw4990** merged commit `c9c337e` into `pingcap:master` on 11 Nov 5 of 6 checks passed [View details](#) [Revert](#)

经过我和 Tidb 几位 reviewer 的共同努力下代码最终被合入到主分支中了。

3、课程反馈

我认为本课程更应该将关注点放在技术上，例如结合具体的项目向同学们演示如何进行项目开源，如何在开源社区中做贡献，一步步进阶。最后是提升本课程的价值如增加对开源精神和意义的讲解，而不是一开始就长篇大论介绍开源的精神价值等，放在具体的项目中进行实践，便于同学们的理解。

4、参考文献

3