

# 《开源软件设计与开发》课程总结

姓名：朱会会

学号：51194507025

## 1、开源理解

在读研究生之前，我并没有接触过 github 和 git，大学的时候老师让我们做系统，也只是百度浏览器搜索并没有想到使用 github，接触 git 也是在开源过程写好代码要提交之后，我们并不知道一次的提交过程是什么样子，所以才想到去了解了一下 git，我也是第一次知道这种协作开发的方式，我之前其实有想过，企业在开发过程中如何将大家的代码汇总，大家又怎么可以方便的看到别人的代码以方便协作开发，后来研究生参与到项目终于知道有 svn 和 git 这些开发协作工具。所以在学习了一些 git 常用的命令之后就开始着手使用 git，我发现它是真的强大，这是一种我们从来接触过的写作方式，版本控制有以下功能：

(1) 协同修改：多个开发人员可以并行不悖的修改服务器端的同一个文件，即最后上传到服务器的是多个人修改后的文件，不会被后传者覆盖。

(2) 数据备份：不仅保存目录和文件的当前状态，还能保存提交的历史状态。

(3) 版本管理：保存每一个版本的文件信息时，一般不保存重复数据，以节约存储空间，提高运行效率。其中，SVN 等集中式版本控制工具采用增量式管理，每次保存有修改的部分，然后跟之前的部分拼接起来；Git 等分布式版本控制工具则对文件系统采用快照的方式，即每次保存文件系统的快照。

(4) 权限控制：能对团队开发人员进行权限控制，比如开发系统的有些部分你只能查看，不能修改；有些部分，你既能查看，也能修改。特别的，Git 所独有的权限控制特点是——能对开发团队外的开发者所提供的代码进行审核，从而管理 Linux 开源社区代码。

(5) 历史记录：可以查看修改的所有信息，包括修改人，修改内容，修改时间……

它还能本地文件恢复到某一历史状态。

(6) 分支管理：Git 的分支管理功能优雅、简洁、流畅，它允许开发团队在工作过程中多条生产线同时推进任务。

## 2、开源贡献

贡献代码这件事说来就真的很有说的了。老师布置的任务是，每个人选择一个开源项目，然后在里面贡献代码。Tidb 选择的人好像是最多的。我也选了 tidb，函数向量化的工作似乎是最容易完成的。但当我们着手想去着手完成作业的时候，一下懵了。第一，我们甚至不知道如何将源代码部署到本地，一个同学教了我们：先用 git 把仓库里面的代码 git clone 到本地，然后下载 go 语言，go 语言使用的 IDE 一般是 Goland，把代码打开，会自动下载依赖，我们照做，总算在本地上部署好项目。第二，我们不知道去修改哪一个包下面的代码，后来经过半天摸索，还是知道了应该在哪里写代码。问题又来了，代码如何修改？go 语言我们没学过，但是看懂代码逻辑还是没问题的，于是我们去看别人的代码，看看传进来的是什么参数，有没有和要写的参数是类似或者相同的，于是仿照别人的代码，我们还是完成了自认为没有毛病的代码。问题又来了，如何在 github 上提交自己的代码？于是，我又学习了一下 git，知道大概的过程是，先用 git 把修改的代码提交到自己的 github 仓库，然后在 github 上提交一个 pr，如果代码没问题的话，就有可能被 merge。最后当我把代码提交上

去的时候，又和 reviewer 进行了一些交涉，他告诉我获取时间用的应该是另一个函数，这个空行要删掉，，，，我照做，当最后我看到他给我的代码打上 LGTM 的标签之后，终于露出了甜美的笑容。我的 pr 也终于被 merge 了。

贡献连接：<https://github.com/pingcap/tidb/pull/13482>

### 3、课程反馈

这节课我觉得老师安排的都很用心，我们在听各路大佬进行授课的过程中拓宽了眼界，认识了开源，同时亲身加入到开源贡献者这一大家庭。但是在我实际的想进行一次代码提交的时候，确实碰到了很多问题，比如一次代码提交的流程可能看起来很弱智，但是对于一个新手小白来说，要摸索很久，所以我的建议是，如果能在上课的时候，具体实操一次简单的代码贡献和提交过程，会让同学们不至于那么迷茫，可能有助于更关注代码本身，而不是花很长时间去苦思冥想怎么去开始动手完成这件事情。

### 4、参考文献

[https://blog.csdn.net/qg\\_41682292/article/details/89456105](https://blog.csdn.net/qg_41682292/article/details/89456105)