

# 《开源软件设计与开发》课程总结

姓名：张洋

学号：51195100040

## 1、开源理解

开源字面意思是开放源代码,但是随着 Git 的发展,开源的含义已经被扩充开放源内容,而这个内容不仅仅是源代码,例如开源电子书 Gitbook,作者可以通过 Git/Github/Markdown 编写协同编写电子书。开源对于避免重复造轮子,传播软件 and 知识有很大的意义。

近几年开源技术快速发展,在云计算、移动互联网、大数据等领域逐渐形成技术主流。开源一方面可以突破技术壁垒推动技术创新,另一方面不可避免的带来知识产权,信息安全等一系列问题,企业如何开源,项目开源流程、开源商业模式、如何使用开源等问题称为产业界关注的焦点。

开源软件市场应用广泛,据 Gartner 调查显示,99%的组织在其 IT 系统中使用了开源软件。开源软件市场规模稳居服务器操作系统首位。根据《Linux 内核开发报告 2017》显示,自从进入 Git 时代(2005 年 2.6.11 发布之后)共有 15637 名开发者为 Linux 内核的开发做出了贡献,这些开发者来自 1513 家公司,全球公有云上运行的负载 90%是 Linux 操作系统,在嵌入式市场的占有率是 62%,而在超算市场占有率达到 99%,智能手机市场占有率 82%。

开源技术的使用方式可以分为三种,第一种是直接使用原生开源技术,第二种是对开源技术进行二次开发再进行使用,第三种是购买基于开源技术的商业软件。企业用户可以再开源技术技术上定制化开发,相比闭源软件,开源软件更灵活,定制化程度更高。使用开源技术,让企业专注于创新。

但是开源也有缺点,开源软件可能具有不够完善的文档,可能由于关键程序员的推出导致开源项目陷于停顿,有的适合开源软件的版本迭代不符号业务需求,只能进行二次开发,导致 Branch 和 Master 的版本差异过大等。虽然目前有大量的开源技术和软件可以使用,但是背后的厂商如谷歌、Facebook、亚马逊、苹果、微软对这些开源技术也掌握着绝对话语权。一旦开源软件的某些厂商利益跟上述公司冲突,不排除被取消软件授权或者相关软件不再更新的可能性。

最近了解到硬件也可以开源,开源软件可以通过定义好的接口来合作,但是硬件领域的软件仿真是不可靠的,只能做参考。硬件的设计修改成本很高,兼容性和灵活性也不足够,但是国外这方面还是做的挺好的如 Arduino、BeagleBone、Raspberry Pi,我相信随着国内开源技术的不断发展,也会有国产的开源硬件产生。

## 2、开源贡献

刚接触 Github 时,发现很多开源的软件,但是很多时候只是使用 Clone, Fork 然后自己修改,在之前项目中用到了一个开源深度学习框架 DarkNet (Star 6k; Fork 10.1k)的 Brach,这个 Brach 是 Darknet Yolo v3 和 v2 神经网络的 Windows 和 Linux 版本(使用了 Tensor Cores),我在 Github 上与这个 Branch 的俄罗斯 Forker 进行交流,提过一个关于 yolo 网络目标检测中 NMS 函数的 Issue:为什么一个的 Object 会有多个不同 Class\_ID 重叠的 Box?这个 Forker 后来回复了我:NMS 仅对相同 Class\_ID 的多个检测有效,如果有自己的需求,可以根据自己需求重写 NMS 函数。

我根据自己的需求修改了 NMS 函数,避免同一个位置出现多个不同类别的预测,在

NMS 函数中添加一条规则：如果存在任意两个不同类别预测框的  $\text{IOU} > \text{thresh}$ ，那么将保留有预测框置信度最高的预测框，将置信度低的预测框删除。以下是修改后的 NMS 函数：

```
void do_nms_sort(detection *dets, int total, int classes, float thresh)
{
    int i, j, k, l;
    k = total - 1;
    for (i = 0; i <= k; ++i) {
        if (dets[i].objectness == 0) {
            detection swap = dets[i];
            dets[i] = dets[k];
            dets[k] = swap;
            --k;
            --i;
        }
    }
    total = k + 1;
    for (k = 0; k < classes; ++k) {
        for (i = 0; i < total; ++i) {
            dets[i].sort_class = k;
        }
        qsort(dets, total, sizeof(detection), nms_comparator_v3);
        for (i = 0; i < total; ++i) {
            if (dets[i].prob[k] == 0) continue;
            box a = dets[i].bbox;
            for (j = i + 1; j < total; ++j) {
                box b = dets[j].bbox;
                if (box_iou(a, b) > thresh) {
                    dets[j].prob[k] = 0;
                    for (l = k + 1; l < classes; l++) {
                        if (dets[i].prob[k] >= dets[j].prob[l]) {
                            dets[j].prob[l] = 0;
                        }
                    }
                    else {
                        dets[i].prob[k] = 0;
                    }
                }
            }
        }
    }
}
```

虽然这个功能没有交 PR，但是我感受到了开源的力量，世界上不同国家不同语言的人可以互相交流，就像数据复用可以降低芯片功耗，代码复用可以降低人脑功耗。

### 3、课程反馈

我觉得可以多讲一下 Git 开源分布式版本控制系统的原理和实践，虽然 Git 很常用，但是好多 Git 的高级用法还没有涉及过。

另外开源项目建议不要局限于几个项目，大家可以根据自己的兴趣加入不同的开源项目进行交流和交 PR。

其他课程内容我觉得挺好的。

### 4、参考文献

[1]中国信息通信研究院.开源治理白皮书[EB/OL]. [www.caict.ac.cn/kxyj/qwfb/ztbq/](http://www.caict.ac.cn/kxyj/qwfb/ztbq/).2018-3.