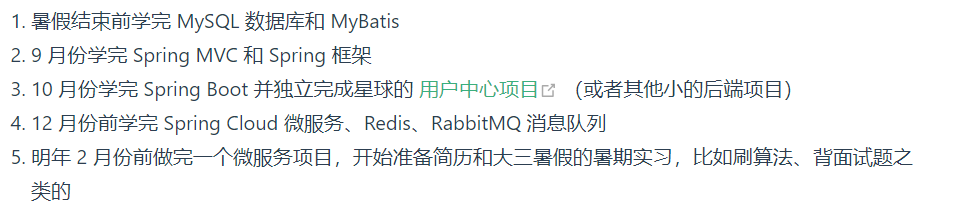
# 一．Plan hou

1. **核心竞争力**
   1. **硬实力：竞赛奖项（算法，软件杯），项目经验**
   2. **软实力：学习能力，沟通能力，团队合作能力，领导力，英语口语**
2. **学习原则：目的导向，主次分明，制作大纲，逐个击破**
3. **项目经验**
   1. **课程作业项目**
   2. **学校研究项目（了解，获取相关的信息）**
   3. **学生组织项目**
   4. **自我发起项目**
      1. **写文档**
      2. **托管到github上**
      3. **让合作伙伴写推荐信**
   5. **竞赛项目**
   6. **实习项目**
   7. **用github跟踪和学习项目**
4. **算法**
   1. **刷板子，转平台，**
   2. **找题库，好的题解**
5. **一些自己无法完成的项目，可以学习相关的东西**
   1. **学会使用GitHub，StackOverflow，慢慢了解相关的博客，开始写博客**
   2. **github上找项目开始学习**
6. Git,Github学习
7. **操作系统，计算机网络**
   1. **操作系统： 重装系统**
      1. **Windows系统，软件生态多，易于玩游戏，喜欢搞升级，升级毛病多，安装编程环境麻烦，可能在关键时刻死机**
      2. **Macos系统：中毒，死机少，mac软件很多不能和win比，软件问题，太贵了，前端开发，苹果客户开发**
      3. **Linux系统：开源，安全，可定制，底层设计，原理可以**
      4. **不知项目，租云服务器(便宜)，xshell运行linux,**
   2. **计算机网络，学习网路知识**
8. **学习前端，后端知识，学会维护网站**
9. **AIGC行业：未来发展路线**
   1. **短视频脚本**
10. **政策相关**
    1. **了解国家政策，以及规划**
    2. **了解行业政策**
    3. **了解学校的一切信息**
11. **了解一些国产AI，和国外的对比，还有一个openAI工作者，手写chatgpt的视频对应源码学习** 
    1. **小模型**
       1. **微软开始制作小模型**
12. **玩转coze，是一个开源平台，可以搭建自己的AI机器人模型**
13. **玩通电脑：**
    1. **解决电脑升级问题 finish**
    2. **解决edge浏览器和chrome浏览器更新问题**
    3. **了解清理电脑的软件**
    4. **重装系统**
    5. **坚持学习ytb对应内容**
14. **self summary;将自己任何时候总结的观点写入记录下来，需要时间形成自己的方法论，不断内化**
15. **iphone与安卓**
    1. **应用生态，软件更好**
16. **Bat种子解决，迅雷下载使用，以后看东西直接下载种子,bat种子下载犯法的**
17. **一些流氓软件，对应解决方案，**
    1. **各种网盘，百度网盘，夸克网盘下载速度缓慢**
18. **获取关于找工作的所有信息，**
    1. **职员内推**
19. **自媒体营销策略**
    1. **自媒体分享渠道，分享一个产品或平台，可以增加在搜索引擎中的排名（SEO）搜索引擎优化**
    2. **增加曝光度，积累信任**
20. **计算机领域的相关信息**

****

# **二.5.31~8.31**

1. **C++深度学习**
   1. **语法**
   2. **进阶：c++ primer ，Effective c++;**
   3. **计算机基础知识：** C++ 语言本身相对更底层，常用于系统开发、嵌入式、高性能计算等领域。因此 C++ 开发者相对于学 Java 或其他方向的同学来说，需要具备更扎实、更深入的计算机基础能力。
      1. **计算机导论（基本概念）**
      2. **数据结构和算法 （每天2小时左右）**
      3. **操作系统 （每天2小时左右）**
      4. **计算机网络（网课）**
      5. **计算机组成原理**
   4. **STL源码剖析**
   5. **Linux服务器编程：**对 C++ 开发来说，Linux 系统是重中之重。掌握 Linux 可以帮你更好地理解系统底层的运行机制，比如进程管理、内存管理、文件系统、网络传输等。
   6. **UNIX网络编程**
   7. **软件开发通用：**除了 Linux 之外，软件开发的其他通用技能都可以融入到日常学习中，而不是花一大段连续时间去专攻。比如企业项目研发流程，读一遍有个印象就好；Git & GitHub 需要在平时做项目、用开源项目时多多使用；23 种主流的设计模式可以每天学 2 - 3 个；软件工程的专业知识大学科班一般都会教，学校不教的话作为一个课外知识去补充即可。
      1. **企业项目开发流程**
      2. **Git & github**
      3. **Linux系统：**
      4. **设计模式：23种设计模式（面向对象，已经或多或少涉及一些）**
      5. **软件工程专业知识：**软件工程强调在软件开发过程中，按照一定的 流程和方法 来组织、规划、管理和控制软件开发过程，以保证开发出高质量、可靠、易维护的软件项目
      6. **软件工程专业学习的好处**
         1. 提高开发效率：软件工程涉及了设计模式、项目框架、工具的知识，学好后可以提高开发效率，缩短软件开发周期。
         2. 提高软件质量：通过一定的流程、方法和工具来规范软件开发过程，可以有效地提高软件质量，降低软件维护成本
         3. 提高团队协作效率：软件工程介绍了一些团队协作工具和方法，可以帮助团队更好的协作，避免开发中的问题
         4. 提高软件安全性：通过学习规范的流程以及软件风险知识，可以培养我们对系统安全性、可靠性和可维护性的意识，从而有效地避免像安全漏洞和程序崩溃等问题。
   8. 后端开发通用：
   9. C++项目实践，求职备战
   10. **涉及JavaWeb的知识**
       1. ****

# **三. 6.31前**

1. 操作系统 （了解如何写一个操作系统）深入学习，结合计算机组成原理
2. 计算机网络 涉及深入，后面还是减少时间细水长流
3. 准备期末考试
   1. 数学相关
   2. 物理相关
   3. 英语相关

# 四.6.31~8.31（可以从github上获取先下载到本地）

1. 每天数据结构和算法（细水长流，english spoken, 英语文献，英语书籍，）
2. 写blog
3. STL源码
4. Linux
5. 设计模式（每天一两个，一共23，卡吗网有，还可以找别的刷题软件，clone到本地了）
6. 软件工程相关知识
7. 了解前后端：后端有C++, Java

# 五.大二

1. 前端相关知识（Java为主），
2. Java学习
3. 后端相关知识

### 知识点

* 数据库
* 开发框架
* 包管理工具
* Redis
* 消息队列
* Nginx
* 微服务
* 容器
* 架构设计

1. python学会写一些小脚本，提高生产力，