[1. 硬件 3](#_Toc515262456)

[1.1 硬件工程师 3](#_Toc515262457)

[1.1.1 PCB工程师、FPGA工程师、IC工程师 3](#_Toc515262458)

[1.2 嵌入式工程师 4](#_Toc515262459)

[1.2.1 系统级别嵌入式 4](#_Toc515262460)

[1.2.2 应用级别嵌入式 5](#_Toc515262461)

[2. 软件 5](#_Toc515262462)

[2.1 桌面程序工程师 5](#_Toc515262463)

[2.2 网络应用工程师 5](#_Toc515262464)

[2.2.1 前端开发 5](#_Toc515262465)

[2.2.2 后端开发 7](#_Toc515262466)

[3. 职位 8](#_Toc515262467)

[3.1 测试工程师 8](#_Toc515262468)

[3.2 安全工程师 8](#_Toc515262469)

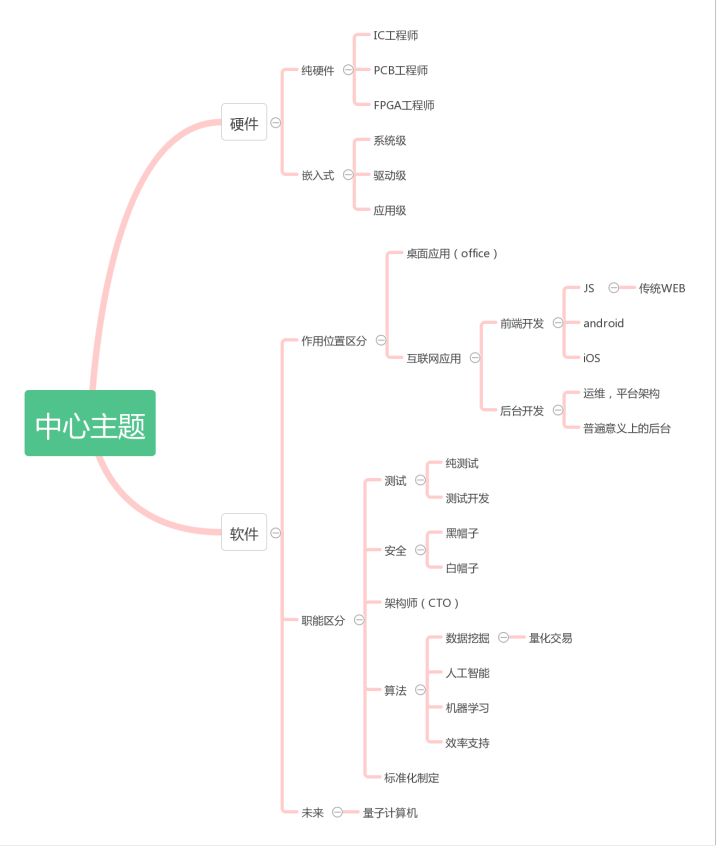
[3.3 架构师 8](#_Toc515262470)

[3.4 算法工程师 9](#_Toc515262471)

[3.5 数据挖掘 9](#_Toc515262472)

[3.6 人工智能 10](#_Toc515262473)

[3.7 深度学习 10](#_Toc515262474)



# 硬件

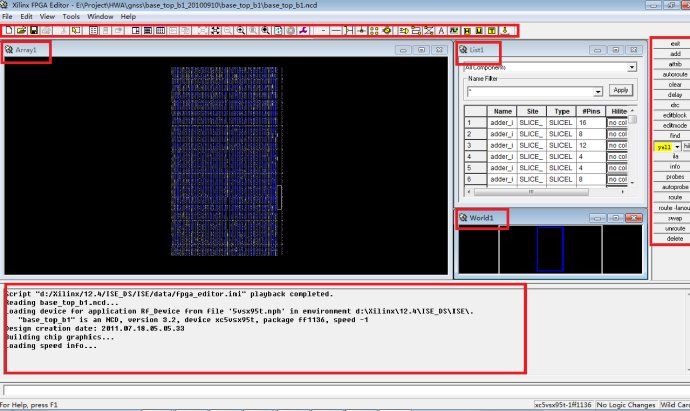
## 硬件工程师

指在计算机领域里，需要接触到电路底层的工程师，实际上在电气领域，在电方向上分为强电和弱电，强电指高压传输，电动机等高电流作为能源动力领域，弱点就是指微电子方向的信息处理领域。计算机硬件工程师的工作范围在弱电，主要内容主要有PCB设计（印制电路板），ic（芯片）设计，FPGA工程师。往上还包括涉及针对特定硬件做系统开发维护的嵌入式工程。

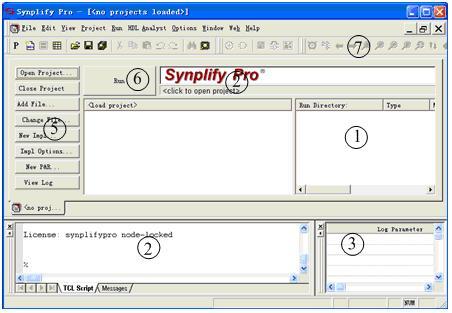
### PCB工程师、FPGA工程师、IC工程师



FPGA全称叫现场可编程逻辑阵列，就是理解为一个可以编程的硬件。传统意义上的集成电路是定制化的，针对特定的需求设计特定的电路板。而fpga可以通过熔断机制来动态修改电路的结构，达到动态编程的效果，简单点说，针对一块FPGA，你可以把他烧制成视频编码器，也可以烧制成无线wifi通信模块。虽然这些算法都可以通过软件实现，但是硬件实现的效果算法成本更低。开发语言常用的有VHDL（硬件描述语言）。



IC工程师就是设计一个高效运算的大脑，涉及到流水线机制，超频，最重要的是解决纳米级别下电流扰动问题，最大的制约因素也是材料工艺问题。



以上三种岗位其实在现代学术分支类还是属于EE下的子类，事实上从事这个方向的人也大部分是学习电子电路工程的同学。这个专业的特点就是强者通吃，全球唯二的两家芯片公司intel、AMD。而且他们的核心竞争力并不是芯片设计能力上，而是在制作工艺上。现在在移动终端统一江湖的还有英国Acorn有限公司设计的ARM芯片，它只是提供一个解决方案，设计图纸，具体的制造还得是华为，三星自己开场设立。但是同时又有很多小众领域，比方说无人机，安全门等等你又需要做专门的PCB设计，所以硬件工程师也是有市场需求基础的，但是也不多。从技能需求来看，要求数学知识扎实，逻辑严密，电子电路知识更多，类似工具反而并不是显得那么重要。从未来的发展前景来看，随着智能硬件物联网的推广，会有一波大红利。

岗位需求：IC << FPGA = PCB

薪资 ：IC >> FPGA = PCB

## 嵌入式工程师

### 系统级别嵌入式

主要是针对特定硬件移植一套操作系统，类似于linux，VxWorks等，甚至于自己开发一个只能满足需求的未知系统，你需要读管脚信息，读时序。但是大部分还是使用成熟的系统移植，自己写系统一来太慢，二来肯定很多坑。相对来说在这个层面上的人都是大牛。或者做一些驱动开发，比如说，每年让无数游戏宅剁手的最新NVIDIA显卡，都需要这个级别的工程师来做相应的显卡驱动。

薪资水平绝对高，起薪百万级别，入门门槛也绝对高，业内最强不是在研究所，而是在公司。

### 应用级别嵌入式

在基于已经能跑的机器上做开发，比如说，操作系统已经提供了最基本的文件管理，内存管理，CPU管理的情况下。你在此基础上做特定应用的开发。普遍打交道的语言工具是C。

做嵌入式应用开发最蛋疼的不是开发而是调试，在这一层做开发会有各种奇葩问题，比方说你昨天还跑得好好的，今天可能就崩了，可能是硬件问题，虚焊？电容击穿？电路板电流扰动？也可能是软件问题，野指针（很少有系统做了虚拟内存，你所有操作都是在实际的内存空间上操作的），内存溢出。

薪水来看，起薪比较低，今年华为给的算是不错了，能到16K，但是随着你对业务的熟悉，工资是没有上限的，最终达到的升级状态就是系统级别嵌入式工程师。

# 软件

## 桌面程序工程师

传统的桌面应用软件类似于office，浏览器啊在PC端上运行的软件。主流的开发框架有Qt，MFC，.NET。现在还比较强势的一些桌面应用领域就是微软的office和桌面游戏公司。以及基于微软的C#的开发应用，基于QT的C++开发应用。其他的不是特别了解，但是给我的感觉就是这个市场不怎么活跃，招人很少，身边也没有认识的人走这条路线。经知友私信补充，这个行业的薪资并没有随着互联网浪潮而有大幅度增长，该知友在.NET平台下使用C#开发，有5年的从业经验，15年的薪资水平在11k左右，后来转行去了android，当然这可能只是个例，有很多厉害的人我们可能不知道，先谢谢这位知友知友了。

## 网络应用工程师

从google发迹以来，基本上所有新兴的巨无霸公司，facebook，bat，amazon都是依赖于互联网的发展，依赖于这群互联网应用工程师。同时他们也就是在网上自称为码农的这个群体。他们这个群体应该能占到计算机领域70%以上的研发人员。

### 前端开发

广义的前端就是指呈现在用户视觉的领域，直接可以让用户感受到的开发，往下会细分为web，andoroid，ios。

#### web前端（h5前端）

web前端指在浏览器（包括手机和pc浏览器，或者是webview控件，甚至于搭载JS引擎的任意环境）上运行的一系列应用。

传统意义上的前端主要是指网页，html,css,js那一套了，或者再加上html5,css3。但是最近几年前端发了很多变化。

这个方向是这几年的大热门，随着机器性能的提高，v8引擎的普及，虽然js和c，java的运算效率还是差上很远，但是目前来看对于呈现展示内容已经基本够用，所以，近几年出现了各种叛逃，比如说cocos2d-js来做游戏，react-native来做客户端啊。同时，随着大量人才的涌入，特别是这两年，大量的前端框架，解决方案让你感觉跟不上时代，讲真，两年前还是JQuery的天下，现在你不知道Vue，Angular都不好意思打招呼。同时由于机器性能的提高和V8引擎的强劲有力，脚本语言的另一个优点就是无须编译运行，导致了它可以实现一个其他语言无法做到的事情---动态更新，可以在app不发版的情况下动态的大幅度更改它，进一步的增加了JS的需求量。

薪资比起前两年来看涨了不少，而且市场需求量也很大，基本上任何应用都没有完全脱离web的框架内。门槛也不算高，但是往下深入一样会有很多东西要学。

#### android和iOS

android是一个基于linux内核开发的开放源代码移动操作系统，由Google成立的Open Handset Alliance（OHA，开放手持设备联盟）持续领导与开发，主要设计用于触屏移动设备如智能手机和平板电脑。iOS（原名 iPhone OS，自 iOS 4 后改名为 iOS）是苹果公司为移动设备所开发的封闭源代码操作系统，所支持的设备包括 iPhone、iPod touch 和 iPad。

目前看来，智能机市场上，android的市场份额在四分之三以上，ios占了大概五分之一，剩下的都是什么塞班，winphone我们应该不用考虑，你应该也不会想做这个的。虽然android和ios市场占有率相差悬殊，但是android工程师和ios工程师基本上能维持相同的人数，所以你如果想从事移动端开发，就业市场来看的话,没有很大的区别。

技术层面来看两个其实比较像，主要的区别其实是生态圈的比较。

相应来说，可能android的入门成本门槛比较低，java相对于Object-C，swift来说还是比较流行的，同时，相对于iOS（mac电脑+苹果手机+一年100刀的开发者账号）的起始1.5W的投入来看，可能android的竞争会更激烈一些。

由于android的开放性，基于开源的linux开发，所以业内会有很多公司针对于android做所谓的“深度定制”，“适合国人的智能手机”等，那就导致了android的各式各样，不同的手机厂商在某些细节方面又有不同的理解，倒是android开发最大的一个问题就是适配。同时由于android的权限开放性，他会有很多可以hack的地方，针对android的木马病毒（360管家）很多，你要考虑到的东西会很多，不过目前来看，android的每次发版，安全性也越来越好，权限管理也越来越规范，以后会越来越好。而iOS的封闭特性，相对来说被发现的漏洞少（并不是不多，我个人觉得绝对比android多），但是每次爆出都是大新闻，它的开发相对来说规范很多，有很多成熟的解决方案，他的主要问题其实是iOS审核这块，大部分的公司都有过app被appstore毙过的经历。

另外，最近苹果公司也在推swift，如果你要做iOS，由于各种历史问题，OC并不能完全弃掉，但是swift也会成为你求职的考点或者亮点，通过同事的反应来看，swift比OC更爽，它更愿意做swift。

### 后端开发

后端开发指的是，在为前端提供数据支撑的一个大的总类，包括数据库，业务逻辑处理，数据处理等。主要的开发语言分两个方向编译型语言，java，c/c++等，他的特点就是运行效率高，相同机器配置的情况下能支撑更大的访问量，适用于超高并发度，比方说淘宝后台是java，网易游戏腾讯游戏是c++。另一种就是以php、python等解释性语言，他们的特点就是开发效率高，无需编译，写完就能运行，主要针对一些运算量不大的中小型网站，比如说你的个人博客，管理后台。

#### 运维工程师

特意把它从后端中拿出来，主要是他的开发任务比较少，但是同样很重要，当数量量达到一定程度，数据库服务器都有上千台的时候，就需要专业的人士做做数据维护的工作了，它的工作难度一样很多，容灾备份，热替换。而且目前的趋势都是服务化，组件化，虚拟化，一样有各种难题有待于你去解决。但是从市场需求来看，只有大厂才会招专业的运维工程师，创业公司100台服务器以下的小公司一般都是托管到各种云，然后由后台工程师兼任。

#### 通俗意义上的后台开发

也就是那些经常说自己是业务狗的屌丝码农，觉得不做架构设计比较low，他们主要的工作就是实现pm的工作需求，比方说双十一来了，pm说，这个要添加一个抢红包的功能，然后后台业务狗就忙起来，要解决高并发下的死锁啊，重复请求啊，带宽啊各种功能，当然也需要前端，但是前端并不需要考虑复杂的并发性问题。而如果这些业务狗在处理高并发问题上有所建树，那么他们就能一步步升级为架构师。那个时候就可以把屌丝码农的屌丝去掉了。

# 职位

## 测试工程师

一个产品或者一个新的功能需要上线，必须要经过完备的测试，测试工程师目前其实有两种，一种是纯测试，另一种是他的升级版测试开发工程师，它能在测试的过程的顺手把小bug改了而不需要返工。好的测试开发工程师工资也可以比做开发的工资高。

而且你们也不要小看测试这个东西，不要以为你只会点点鼠标看看对不对这么简单，合格的测试工程师需要熟悉各种测试工具，能自己写测试脚本，能找到bug，而且还能知道为什么会出现bug。这个工种一般也都是大公司才会配置的，对于小公司来说，一般就是开发人员自己也是自己的测试人员。

## 架构师

从这里往后，已经算是程序员界的高富帅了，可以称的上是半个科学家。架构师的工作差不多已经基本脱离代码了，他的主要工具从VIM，Eclipse变成了word，ppt。他主要觉得技术选型，针对大容量高并发的问题制定解决方案。基本上他们都是由后台工程师升级打怪升上来的，好像听闻业界也有前端工程师升级为了CTO做架构的，但是毕竟还是少数，node的效率还是差了很多。

## 算法工程师

这类人，基本都是大学参加ACM竞赛拿过奖，或者本身思维逻辑严密性高，数学基础扎实，算法牛逼。如果你是一个数学系或者物理系的优秀毕业生想转IT，这个可以作为你的主要方向。这个算是建筑系里的学院派，从学校毕业之后就直接达到了这个等级，起薪很高。

## 数据挖掘

它的作用就是从一堆数据中挖去你你想要的信息。打个比方，给你2008年到2016年所有的房屋成交量，房屋成交价格，土地成交量，人口增长，银行贷款利率变化等等很多信息，让你预测明年房价变化情况。他们就是干这个的，这个工作分三步走，爬数据，建模，写算法验证。

爬虫工程师，就是从各种国家统计局网站，链家网站爬去数据，存储为结构化的数据。

数学建模，建立各种数据之间的影响关系，影响因子。

写算法验证，大部分数据挖掘工程师好像都是用python，工具比较多，语法简单。

难点其实不是在写代码部分，而在于建模，如果设计一个有效的模型来量化这些数据间的联系。其中的影响因子权值就是他们的生命价值所在，所有他们会光荣的把自己称呼为“调参侠”，顾名思义，就是天天调整参数，让模型运算结果来匹配实际情况。

薪水很高，一般统计分析局或者大公司才会招人，而且还可以独立创业，像是知乎很有名的团支书，拿数据打脸的那种牛逼轰轰的人物。

另外再而外介绍一下量化交易，量化交易是指以先进的数学模型替代人为的主观判断，利用计算机技术从庞大的历史数据中海选能带来超额收益的多种“大概率”事件以制定策略，极大地减少了投资者情绪波动的影响，避免在市场极度狂热或悲观的情况下作出非理性的投资决策。这个是计算机屌丝和金融高富帅融合的比较成功的一个新的工种，通过一个优秀的交易员建立一个理论交易模型，再有程序员实现代码，这个算是数据挖掘领域比较直接的变现交易。

## 人工智能

这方面很多人可能理解有误。经常看到的siri，或者聊天机器人之类的都是属于这种东西。他的本质还是一个有限自动状态机，在一个状态下在很多因素的影响下走向下一个状态，理论上你把你在三维世界所有的输入都考虑进去，而且都分别定义了相应的输出，他是可以做到人工智能的。但是这样写出来程序会非常大，电脑跑不动。我个人的理解，基于计算机简单的01唯一向性，真正意义上的人工智能还是蛮难实现的。比如说之前非常火的李世石大战阿法狗，在这种特定的应用下，输入很简单，围棋上的落子16\*16，状态也很简单，2的16\*16次方，在这个复杂度下，肯定也是没办法枚举的，他们会有降维算法各种算法来聚类他们的状态。

## 深度学习

我们上面所说的数据挖掘会有一个调整参数的过程，而在深度学习领域，他会添加一个正向反馈，让你的参数由机器自动调整，让他计算的结果去自动匹配实际的结果。这个就是所谓的机器学习，就是把人为调参的过程让机器自动完成。

此外，人工智能，深度学习，数据挖掘并不是分开来看的，他们会存在一定程度上的重合。他们其实真正的难点实在各种降维算法，概率论，退火等数学领域。所有说他们算是半个科学家也就是这样原因了。