

科学地提问读后感

在阅读了[提问的智慧](#)和[别像弱智一向提问](#)之后，本人对提问的艺术有了一些思考，联想到两三年前刚进入大学时那种无知的样子，又一次被自己逗笑了。接下来就是我的具体感想。

关于提问前的流程

在向大佬进行提问之前，要确保自己思考过并且尝试解决过问题，一般来说没有经过思考的提问是不会得到解答的。那么如何尝试解决问题呢？

1. **STFW(Search The F**king Web)**: google、bing、百度(虽然它很垃圾)都是我们的好朋友，在发现问题时最无脑的解决方法就是将报错信息 copy 至搜索引擎，如果前人有人在网络上发布过相关问题并且得到解答，那么自己的问题一般也会解决。这让我想到了大一程序设计期间，遇到 bug 先找老师，结果老师说：“你去百度啊！！”，现在想起来，老师说的不无道理。
2. **RTFM(Read The F**king Manual)**：相比起搜索引擎，官方文档会更加权威和正确。但是阅读官方文档需要带着脑子思考，因为文档涵盖内容很多，需要思考该问题对应文档中的哪个部分，以及根据文档内容思考问题出现的原因。
3. **RTFSC(Read The F**king Source Code)**：阅读源码也许是个困难的事情，但往往也是最有效的、最能帮助成长的，通过阅读源码会帮助了解程序运行的原理，这个是从事互联网行业必备的技能。在半年前参加过 OpenDB 数据库的内核开发大赛，其实就是通过读源码举一反三开发出新的功能，可见阅读源码有多重要。

以上便是在提问前自己需要做的事情。

关于提问的艺术

我很喜欢文中的一句话：“In the world of hackers, the kind of answers you get to your technical questions depends as much on the way you ask the questions as on the difficulty of developing the answer.”(大致翻译：在黑客的世界里，答案的好坏，取决于提问方式和问题难度)。

所以如何向大佬提问也是一门学问。

1. 礼貌

连小学生都知道与人交流要礼貌，在提问的时候加上“请”、“谢谢”等语句会让被提问者倍感亲切。

2. 合适的场合

需要在合适的地方发布提问（总不能在弱智吧让大家解决程序编译的问题吧）

3. 聪明的提问

聪明的提问是最重要的，需要在问题中说明出现了什么问题、设备环境的信息、报错的日志信息、有可能是什么原因导致的、使用了哪些解决方法等，加上这些元素会让问题变得有意义起来。

从提问向不提问转变

未来在进入职场后，很少会有人为你解决工作上的问题，所以在大学期间需要提升独立解决问题的能力。也许在刚入门时需要经常向大佬请教问题，这是没错的，但是自身也需要慢慢调整，要学会三句箴言（STFW、

RTFM、RTFSC)，努力地跳出舒适圈，如果能做到不向他人询问就能解决问题，那么就会成为一个“成功的 man”！！

关于预学习阶段

在预学习阶段，我遇到了不少问题。

1. 在完成复习 C 语言时，我遇到了有关编译的问题，出现问题的主要原因是我没有弄清楚静态库的编译顺序，最后导致了程序中的函数未定义。当时为了解决这个问题，我将链接的静态库改为了动态库，最后编译成功。但是，依然没有解决为什么链接静态库就无法编译，后来我通过 STFW 知道了链接静态库有顺序上的要求。
2. 在完成 PA1 时，阅读源代码着实很痛苦，为了了解其运行原理，需要从 main 函数开始一点点跳转阅读，最后也算是清楚了 nemu 调试器的原理。
3. 在学习数电时，由于我先前没有了解过 verilog，而且受程序设计的影响，很容易就开始行为建模，通过一次预学习交流会，老师和大佬们很耐心的给我提出了建议，也给了我几个学习 verilog 的网站，目前还在培养数电思维当中...