Be a Hacker, 从0到1，构建自己的操作系统

在股市大跌前一段时间，我熟悉了一个词叫“杠杆”,它是指通过某种机制放大你的资金效益，10块钱能当一百块用，赚能赚十倍，同理，亏损也会放大10倍。后来我意识到，人的学习成长，其实也应该使用杠杆效益，例如，如果你掌握了编译原理，那相当于你掌握了所有的编程语言,无论java,python,c++,javascript等逃脱不了编译原理的范畴（特此推荐我的课程：用java开发编译器）.同理，如果我们学会做一个操作系统，我们就可以一下子掌握硬件，计算机体系结构，复杂的软件架构，各种算法与数据结构，大规模复杂软件的开发模式，通过开发一个操作系统，你就能掌握任何复杂软件系统的构建能力，可见，学习开发操作系统，其杠杆效益是非常明显的。

当然，收获与付出是相对的，要想获得上面的收益，我们不得不付出一些成本，那就是陡峭的学习曲线。为了回避复杂性，大学课本往往以务虚的方式讲解各种操作系统原理和算法，就像说大道理，头头是道，但你根本无法实践，另一种是“情景分析”，将操作系统的某个局部做切面后进行讲解，但现代操作系统异常复杂，当前光linux的内核代码就有三亿多行，在不了解上下文的情况下，单纯的去吃透局部，不太现实。如果把linux当成亚马逊丛林，那么各种情景分析就类似于给你各部分的局部地图，让你按图索骥的走出去，想必这不大可能吧。

本课程的目的是遵从人的认知方式，由感性到理性，由小到大，由浅入深，从一个最简单的引导扇区开始，逐步增添代码，最后形成一个拥有GUI, 能够运行多种程序的多任务操作系统，在本课程的第一节有一个SampleOS演示，大家可以看看，我们最终完成的操作系统样子（我们要做的更好）。

众所周知，操作系统的开发及其复杂，主要是它作为平台，承担的是做为软硬件桥梁的功能，必然比一般软件更难理解，同时核心部分除了各种复杂算法外，还包括对cpu，内存，IO等硬件操作，各种逻辑交叉关联，从而显得很晦涩。现在有不少开源系统，但代码往往上万乃至几十几百万行，而且相关文档要么缺失要么语焉不详，一个没经验的初学者要求理清其中头绪，几无可能。

我曾经遵循上面说到的各种方法去学习实践操作系统，但都中途而废，铩羽而归，直到经过大约十年的工作，学习，研究，实践，如今才初步掌握了一个操作系统的开发脉络。我相信在学习路上，不止我一个人遭受过这种挫败感的打击，因此我想开此课程，让大家不再承担这样的挫折，同时想将那种通过努力完成一个复杂软件系统的成就感和自豪感传达给大家，让大家享受的编程的乐趣，是我的最终使命。

本课程需要一定的汇编语言和算法数据结构基础，但只要你愿意坚持，这些都不是问题。

接下来，我们立马看看我们要做的”样板房”操作系统，以及我们如何开始第一步，操作系统开发的第一步,用java开发一个Hello World操作系统内核.以下是描述具体技术细节的博客链接：

<http://blog.csdn.net/tyler_download/article/details/51729141>