[如何学习Windows编程](http://blog.csdn.net/oldmanzhao/article/details/9387459)

标签： [Windows C++](http://www.csdn.net/tag/Windows%20C%2b%2b)

2013-07-20 01:57 18264人阅读 [评论](http://blog.csdn.net/oldmanzhao/article/details/9387459#comments)(14) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/oldmanzhao/article/details/9387459#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

有很多初入门的小朋友跟我说，他们想学“底层”的技术，因为觉得技术含量高，所以想学Windows C++开发，看着调用一个个的参数超级多，调用起来超级麻烦的Windows API，觉着很酷。于是我念头一转，就有了这篇文章，根据我多年来做Windows 客户端的经验，跟大家谈谈Windows C++，该从哪一方面入手，这不会是一个有技术深度的文章，只是偶尔会涉及到一些术语以及一些入门书籍而已，对于Windows开发的老鸟们来说，只当是看着好玩了。

首先来谈谈最初的Windows开发。

如果打开《Windows程序设计》这本书，关于Windows开发的来龙去脉就一目了然了。但是要注意，《Windows程序设计》是一本纯讲述Windows API的书，确切的说，这本书主要讲的是关于Windows 用户态的API，也就是User32.dll GDI32.dll这两个库文件中的函数，大部分的内容都是关于Windows最基本的窗口元素的操控方面的API，比如创建窗口，绘制窗口，关闭窗口，消息循环等等。这本书不需要你有C++方面的知识，理论上说只要你有[**C语言**](http://lib.csdn.net/base/c)的基础就可以看懂，但是如果你连C语言的基础也没有，看这本书可能会发生一些困难，如果你学过其它一门非面向对象的语言，这本书实际上也能够勉强看懂。我在大学的时候就看了这本书，但是直到现在也没有看完，书中的某些知识，实际上在Windows开发中已经很少用到了，我个人认为是没有必要把这本书从头到位看一遍的，在看了前面几章以后，我就蠢蠢欲动，写了一些简单的程序，迅速巩固了关于这些API调用的记忆，为今后进化到Windows C++阶段打下了一个良好的基础。所以，写程序是很重要的，这个阶段，我建议大家写一些简单的程序，比如打字测速程序，自己创建一个消息循环，创建一个窗口，创建一个光标，用GDI函数输出图形及文字，这样下来，基本的Windows程序设计概念就有了，不过注意写的时候不要大段的Copy书上的代码，全部自己写，否则无助于理解各个API的含义，这个程序，因为是自己写的第一个Windows程序，所以时间限定在一个月到一个半月之内比较合适。度过这个阶段以后，Windows编程的基本概念就应该在你脑海中形成了。

下面进入第二个阶段，也就是C++风格的Windows开发，也就是职场上需求量很大的Windows C++程序员干的事。

这时候你需要了解一下这个老掉牙的叫做MFC的东西，MFC是啥呢，其实是一个很原始的，用代码来拼搭界面的C++库，当然，因为是C++库，所以你必须具备了一定的C++知识。不过别因为原始就小巧了它，MFC比传统的Windows SDK开发方式还是要前进了一大步的，因为不管怎么说，它总算沾了面向对象的一点边，有那么一点点面向对象的意思了，比如一个窗口，会被封装成CWnd类，里面有创建窗口的成员函数，窗口绘制的函数也被封装了进去，CWnd又会有其它的窗口派生类，比如CView啊CFrameWnd以及各类窗口控件。学了MFC，你就基本上对Windows C++开发入门了，换而言之，如果你学的还算过关的话，应该能在市场上淘到一份关于Windows C++开发的工作了。其实现在很多公司还在使用MFC这个老黄牛，虽然MFC跟现在流行的界面库的易用程度比实在是差的太远，但是MFC的界面效率高，细节操控丰富这仍然是不争的事实。

在这个阶段上，你可以做一些简单的稍微上得了台面的程序了，比如开发一个基于IE内核的浏览器，之所以我让你开发这个程序，是因为开发IE相关的程序能学到很多跟COM有关的知识，COM也同样是Windows客户端开发人员的必备技能，它的生命力跟MFC一样的顽强，甚至有过之而无不及。

这个阶段上，如果你对MFC足够熟悉了，还可以接触一下WTL这个轻量级的模板界面库，其实如果有了MFC的基础，WTL的学习过程是很愉快的，很多MFC中的概念，比如消息映射，在WTL中都有对应的东西，WTL更小巧，封装的更高端（模板库嘛），但是微软官方不支持，所以不像MFC在MSDN上有专门的文档。最重要的是，WTL和ATL可以结合的非常好，这样就使得在WTL上调用ATL写的COM组件或是在ATL写的COM组件中调用WTL的窗口变的简单轻松，所以，以我现在的脾气，是更加推荐大家使用WTL而非MFC的，但是毫无疑问的，WTL的学习成本更高，所以仁者见仁智者见智啦。

接下来就该学习COM了，学习COM嘛，几本关于COM的基础理论书籍还是要看的，比如《COM技术内幕》这个本书我比较推荐，比较浅显，适合COM的初学者，另外潘爱民的《COM原理与应用》我也看过，马马虎虎也还算过得去吧，虽然感觉还是前面那本更好一点。至于那本神乎其神的《COM本质论》，我觉得不看也罢，那本书初看的时候没看几页就走神，实在是晦涩，素我直言，《COM本质论》这本书，如果你没有好的C++基础还是别看了，纯粹给自己添堵。

等到有了COM的基础概念了，就可以看看《深入解析ATL》，ATL就是用来写COM的，虽然它也封装了一些窗口，控件，不过这本书稍微写的深了一点，大段的代码剖析，初学者看起来还是有难度的，我当初很多地方都是强迫自己看完的，看完了后，收获还是不小的。

在你有了一定的Windows程序开发经验之后，可以看《Windows核心编程》，这本书讲的都是关于Windows内核方面的东西，比如进程啦，内核对象啦，线程啦，DLL啦，SEH啦，总之我的看法，这本书与《Windows程序设计》遥相呼应，弥补了前者的不足，大篇幅的讲述了Kernel32.dll里面的函数，这本书很多内容我都通读了好几遍，某个国内排名前三的互联网公司面试，面试官的很多问题都是这本书里的东西，所以强烈建议大家把这本书好好学习一下。

总结了一下，看书是很重要的，但是不要忘了写代码，我得感觉，看书的时间和写代码的时间在三七开是比较好的，战争中学习战争，实践出真知，这才是最重要的。