

编程高手必备的思维模式：为什么抽象在计算机科学中如此重要

Original 码农的荒岛求生 码农的荒岛求生 2020-08-07 08:15

计算科学中的所有问题都可以通过增加一层抽象来解决。

All problems in computer science can be solved by another level of indirection.

没有抽象的世界

想象这样一种场景，如果我们的语言中没有代词这种形式，那么我们想表达“张三是个好人”该怎么说呢？可能是这样的：

“你还记得我说过的人吧，穿着邋邋遢遢的，公司在中关村，整天背着个双肩包，写代码的，天天996，这个人是个好人”，看到了吧，在没有代词的情况下我们想表达一件事是非常困难的，因为我们需要**具体的描述清楚所有细节**，但是有了“张三”这种抽象后，一切都简单了，我们**只需要针对张三这种抽象进行交流**，再也不需要针对一堆细节进行交流了，抽象大大增强了表现力，这就是抽象的力量。

接下来回到计算机世界。

计算机使用层面

我们在使用计算机时其实抽象就在发挥作用，在Word中编辑文档时我们不会去考虑CPU是如何处理这些字符的，这些字符是如何被保存到磁盘的。在浏览网页时我们不需要关心网页中的数据是如何在网络中传输的、浏览器是怎样把这些数据适当的渲染出来的，我们需要做的仅仅就是在Word中简单的输入字符，用鼠标或者手指滑动网页。

因此只要在使用计算机，那么抽象就在发挥作用，只不过是我们没有意识到而已，而之所以我们没有意识到是因为抽象工作的太好了。

编程语言层面

程序员也可以从抽象中获得极大好处，因为软件是复杂的，但程序员可以通过抽象来控制复杂度，方法就是抽象。

比如一个好的设计就是对某项功能抽象出一组简单的API，这样其它程序员在使用这个模块时只需要关注这几个简单的API而不是一堆内部实现细节。

不同的编程语言提供了不同的机制来让程序员实现这种抽象。

比如面向对象语言(OOP)的一大优势就是让程序员方便进行抽象，这样类的使用者就无需关心类的实现了，更不用提OOP中的多态、抽象类等，有了这些程序员可以只**针对抽象而不是具体实现进行编程**，这样的程序会有更好的可扩展性，也能更好的应对需求的变化。

系统设计层面

计算机从本质上讲就是在抽象的基础上建立起来的。**计算机科学中的一大主题其实就是在不同层面提供抽象表示从而对外屏蔽实现细节。**

对于CPU来说，其对外提供的是一堆指令集，程序员只需要使用这些指令就可以指挥CPU工作了，这样就无需从细节上知道CPU是如何取出指令、执行指令的。

在操作系统层面，我们将I/O设备抽象成了文件、把程序的运行抽象成了进程、把程序运行时占用的内存抽象成了虚拟内存、又把进程和进程运行以来的环境抽象成了容器、最后把所有的一切包括操作系统、进程、CPU、内存、磁盘、网络抽象成了虚拟机。现在虚拟机技术是云计算的基石，实际上这种技术在上世纪60年代就出现了，并在当前火热的云计算中大放异彩。

难怪计算机科学中有一句名言，“计算机科学中没有什么是不能通过增加一层抽象解决的”，当然后面还有半句，“除了存在太多抽象层这个问题”。

总结

抽象的目的其实就是通过移除不必要的信息从而减少复杂度，因此抽象可以让我们更加关注重点。

在这里没有用太多编程语言中的示例来讲解，其实这也是一种抽象，那就是学习编程也好其它领域也罢，最好理解“顶层原理”，这个是通用的。个人一直有这样一种观点，那就是学计算机不是学一堆语言语法，那不过就是一些文法规则的实现细节罢了，编程高手之所以是高手不在于比别人有多了解一门编程语言有什么稀奇古怪的用法，更重要的还是这所说的“顶层原理”，那么顶层原理存在哪里呢？就存在于我们常说的基础中：**操作系统**、编译原理、网络、数据结构算法等。

最后祝大家周末愉快！

People who liked this content also liked

一年赚了100多万，美金！

码农的荒岛求生

