

第 6 讲 由机器语言到高级语言-程序编写编译

1、快速浏览---本讲视频都讲了什么？

【视频 6.1 由机器语言到高级语言】

由机器语言到高级语言，是计算机语言的一个发展过程。高级语言是怎么提出的，它与机器语言是一种什么关系，课程视频用一个简单示例揭示了它们之间的关系，……。

【视频 6.2 高级语言(程序)的基本构成要素】

所有的高级语言程序都有一些共性的构成要素，比如常量与变量，三种类型的表达式，赋值、分支控制和循环控制语句等。课程视频 6.2 介绍了这些共性概念，并给出了一些常用结构程序的执行过程模拟，具体的计算机语言，如 C 语言、Python 语言，只是对这些共性要素表达方式上有差异，而理解了这些共性构成要素后再学习具体的计算机语言就是容易的事情了，……。

【视频 6.3 用高级语言构造程序】

程序是构造出来的，通过函数对一组语句的封装，可以一层层构造程序。**注：结合着第 3 讲内容的理解，函数是将一组语句用一个名字来代替。**既可以自顶向下的构造，又可以自底向上的构造。视频中还简要介绍了程序开发环境。

【视频 6.4 计算机语言的发展】

本段视频展现了高级语言的发展历程。各种计算机语言是如何提出的，提出新的计算机语言的思想是什么，怎样将用新提出语言编写的程序转变成可执行的程序，能否提出新语言呢？

计算机技术的发展，伴随着计算机语言的发展。现在提出新语言已不是难事，可以站在巨人的肩膀上……。

【视频 6.5 不同抽象层级的计算机-虚拟机器】

本段视频揭示了从不同层次用户看到的计算机是怎样的。将来做什么与计算机相关的工作，就需要学习到什么程度……。

【视频 6.6 计算机语言与编译器--一种抽象-自动化机制示意】

此段课程视频被列为选学内容。本段视频用一个示例，揭示并展现了将高级语言程序翻译成机器语言程序的过程，目的是让你了解而不是让你掌握，如要达到掌握的程度则需要进一步学习。**计算机语言和编译器**是最重要的抽象和自动化机制，计算机语言属于“抽象”，是一种表达程序的手段，而编译器则属于“自动化”是一种**将高级语言程序翻译成机器语言程序**的程序。能否理解“编译器”的思想(不要求你会做一个编译器)，是衡量你是否成为高水平程序员的标志之一。

【视频 6.7 协议与编解码器--另一种抽象-自动化机制示例】

此段课程视频被列为选学内容。本段视频用几个示例，揭示并展现了协议与编解码器的机制，目的是让你了解而不是让你掌握，如要达到掌握的程度则需要进一步学习。**协议**和广义的**编解码器**也是最重要的抽象和自动化机制，协议属于“抽象”，是一种表达约定的手段，而广义的编解码器则属于“自动化”是一种**按协议完成相应工作**的软件或硬件。能否理解“协议”的思想，也是衡量你是否成为高水平程序员的标志之一。因为，比如计算机网络课程就是讲各种各样的协议，协议是计算机与计算机之间交流的手段，也是物体和物体、计算机之间交流的手段。只有深入理解了协议，才能理解各种电子设备、传感设备等。

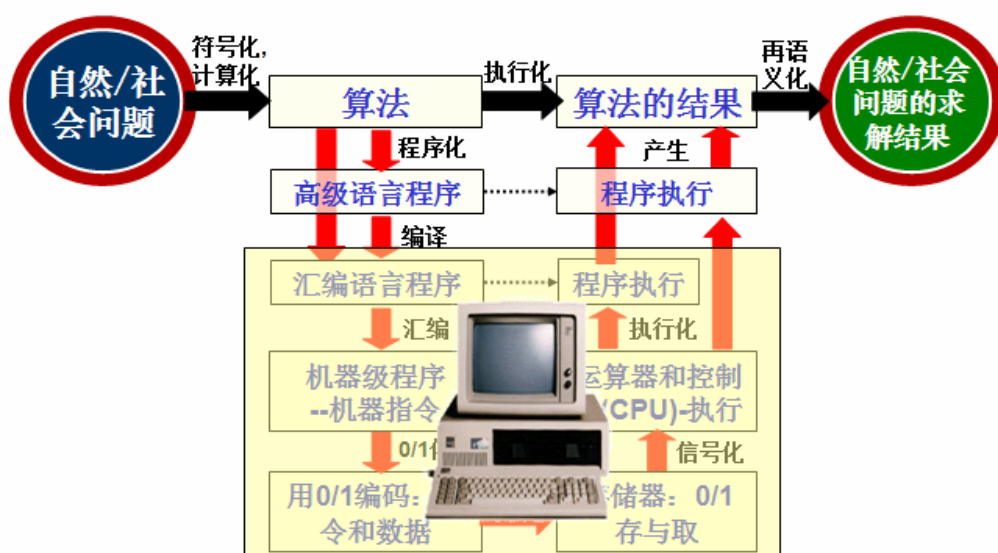
【视频 6.8 分层次抽象-自动化机制示例--操作系统对设备的分层次管理】

此段课程视频被列为选学内容。本段视频以操作系统如何管理设备为例，给大家展现了分层次处理问题的基本思维。分层次抽象是计算机学科最重要的思维方式，如何分层？每一层做什么怎么表达？层与层之间的关系是怎样的？请看视频 6.8。

2、学习要点指南

2.1 要点一：理解“高级语言程序是如何被执行的，以及高级语言的发展与变迁”

学习本讲课程视频，重点在理解“高级语言程序是如何被执行的，以及高级语言的发展与变迁”，即深入理解下面这张图。



第 4 讲讲了“存储在内存中的机器级程序是如何被执行的”，第 5 讲讲了“存储在外存中的机

器级程序是如何在操作系统的管理下被执行的”。而现在一般都用高级语言编写程序，那么高级语言程序又是如何被执行的呢---通过编译器将高级语言程序翻译成机器语言程序，然后就可以被执行了---这也是这三讲之间的联系，自底向上，自简单到复杂的理解计算机是如何执行程序。

有了这三讲的思维基础，再学习算法和程序设计就相对容易了，它也为我们第 7-9 讲算法思维的学习奠定了很好的思维基础。

2.2 要点二：理解程序的基本构成要素，并要能够读得懂一些程序

本讲的另一个要点是理解程序的基本构成要素：常量/变量、三种类型的表达式、语句(赋值语句、分支语句和循环语句)和函数。不同的计算机语言之间的异同是：都存在这些要素，只是书写格式可能不同。书写格式不同，多书写几遍就能掌握。而这些要素的灵活运用，则是程序设计。学习程序设计不在于学，而在于练，多练习读程序，多练习编程序，“练”是硬道理。学好程序设计的一个窍门是“自己将自己当做是计算机，仿照计算机来模拟一行行的执行程序”，你一定能学好！

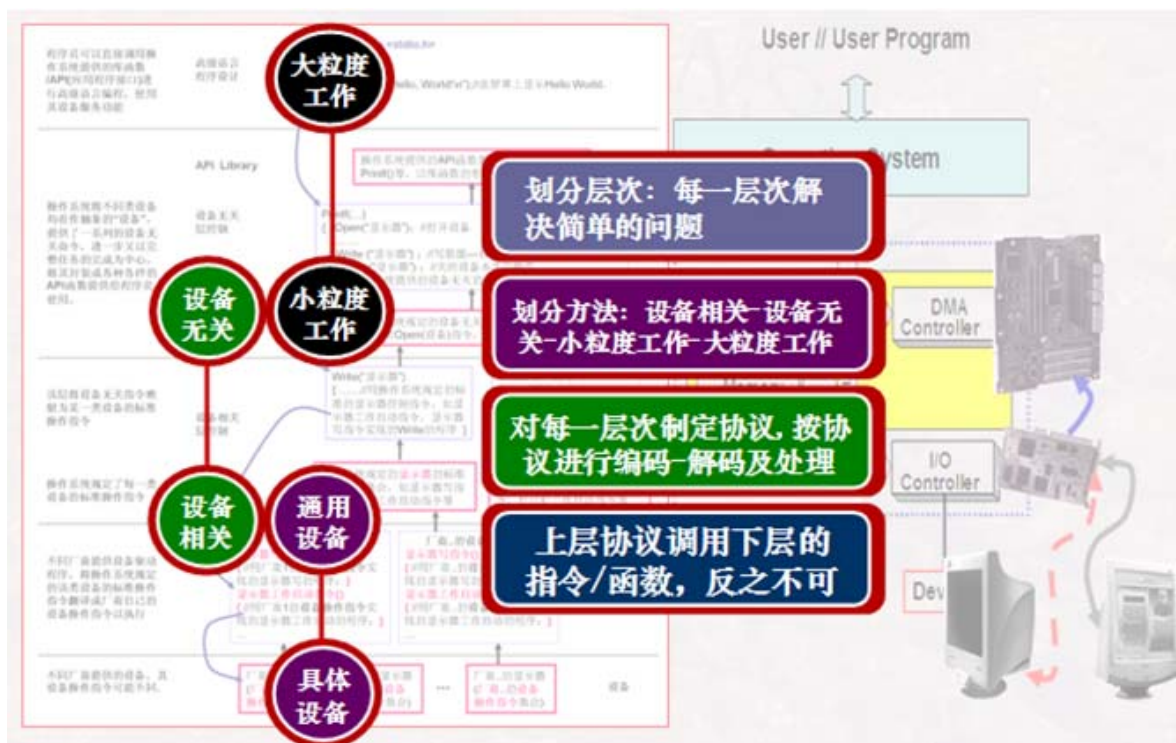
2.3 要点三：深入理解抽象-自动化及其分层处理机制

抽象-自动化-分层次就是计算机学科最本质的东西。

语言-编译器机制解决了人与计算机之间的交互问题---计算机语言是人和计算机都能理解的语言，人用计算机语言编写程序告诉计算机做什么，编译器将用计算机语言编写的程序翻译成机器能够执行的程序。**语言是最典型的“抽象”--它将复杂的千变万化的需求，抽象成基本构成要素，而这些基本构成要素的组合就是程序。**理解编译器的思维，就可以深入理解语言的“抽象”。

协议-编解码器机制解决了计算机-计算机之间的交互问题，进一步解决了物体-计算机、物体-物体之间的交互问题(物联网)---双方在交互的时候要遵循一些约定，即协议，协议也是最典型的“抽象”，依据协议，双方要有广义的编解码器，发送方按“协议”由编码器进行编码，接收方按“协议”由解码器进行解码，广义的编解码器是“协议”的自动化实现。

抽象-自动化，可以分层次来实现。计算机学科充满了分层次抽象的案例：整个计算机网络就是最典型的分层次协议的代表(OSI 七层网络协议，TCP/IP 分层网络协议)，软件的分层次架构，计算机硬件的分层构造与构造集成等等。分层次抽象时，如何分层是最关键的，视频中以操作系统对设备的管理为例讲解了如何分层，如下图所示，理解下图是关键之关键。



3、常见问题

略。