



下载APP



“屠龙之秘” | 实现计算机语言这样的技术能用在哪儿？（一）

2021-10-06 宫文学

《手把手带你写一门编程语言》

课程介绍 >



讲述：宫文学

时长 18:50 大小 17.26M



你好，我是宫文学。

在学习这么多硬知识的间隙里，我给你准备了一些相对轻松一点的内容，想让你转换一下，让大脑休息休息。

不知道你在学习这门课的时候，有没有这样的困惑：**实现计算机语言这样的“屠龙技”，到底在哪儿能够发挥作用呢？**毕竟，不是每个人都有机会成为像 Java、JavaScript 这样的通用语言的发明者的。



我的答案是：其实在任何软件领域，只要你做得足够深，其实都能用上这门课程的知识。我也一直在给你传达一个理念：**任何好的软件，其实到最后都会变成一个开发平台，所以也就需要用到实现计算机语言的这些相关技术。**

所以，在这里，我想分成两节课或者更多节课，给你介绍一些有意思的应用场景，希望你能从中得到启发，在自己的工作中也能更好地运用我们这节课的知识，做出一些耀眼的成绩。

今天这节课，我会跟你分享自动化控制领域（或者简称自控领域），对软件编程的需求，以及如何设计一个开发平台来满足这样的需求。在学习这节课的过程里，你可以不断地做一些印证，想想你可以用在课程学到的知识，怎么来满足这个领域的需求。

但是，我需要补充一句，我自己并不是自控领域的从业者。只不过，我恰好有一些朋友，是国内这个领域顶级的、资深的专家，所以我就有机会学习到了这个领域的一点知识。并且，我跟他们参与了一个工业领域的操作系统项目的策划工作，这也激发了我对这个领域的编程模式的兴趣。但是呢，我对这个领域了解得并不那么深，也不那么专业，我主要关注其中与编程技术有关的部分。

那么首先，我们来了解一下自控系统的作用和应用场景，方便我们理解它对编程技术的需求。

自控系统的应用场景

计算机技术在工业领域有着丰富的应用，其中一个应用领域，就是自动化控制。

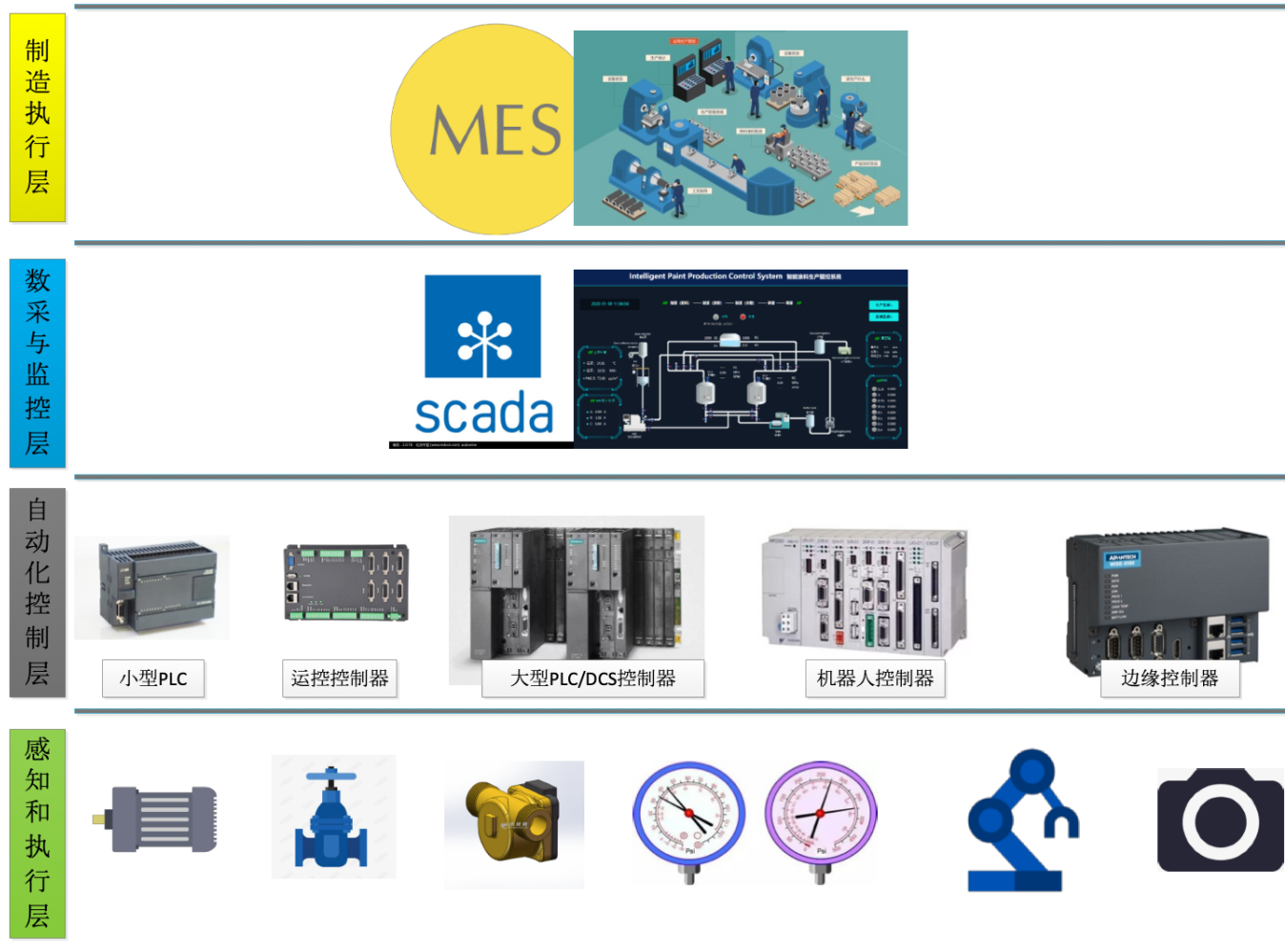
其实自动化控制离我们特别近，你肯定不陌生。比如，当你乘坐地铁或高铁的时候，它们总是能够准确地停在站台边，停靠的位置甚至可以精准到厘米级。这里面就有自动控制的程序在起作用。

自动化控制也有在工厂里的应用，你可能没那么熟悉，但也跟你的生活密切相关。比如在发电厂里，我们需要对发电机组进行精确地控制。如果转子的旋转速度出了问题，那就会酿成巨大的灾难。在核电厂里，我们也需要对核燃料进行精确地控制，以便正常发电，不要引起事故。

你知道，制造业是一个现代国家的根本。而现代工厂的生产过程，几乎全部都会用到自动控制系统，包括离散制造型企业，就是给我们制造手机、电脑、汽车的企业；还有流程型企业，比如各种化工厂，还有前面提到的电厂。

最后，其实我们身边应用得非常广泛的物联网，也属于广义上的自控领域。比如，你家里的智能电表、智能门锁，还有小区停车场的自动道闸等等。

所有这些应用场景，基本上都会采用类似下面的这种技术架构：



你看这张图，最底下是各种生产设备。在这层之上呢，是一些自动化的控制器。控制器最重要的作用就是控制设备的运行。它的原理是这样的：设备在运行过程中，会不断产生信号，信号会被实时传输给控制器。控制器接收到信息以后，在内部进行实时计算，然后根据需要输出一些控制信号，让设备的运行做出一些改变。

通常来讲，控制器对计算能力的要求不会太高，控制的逻辑通常也并不是特别复杂，因为控制程序的时延必须很低。这也很容易理解，比如你开车的时候，肯定不能容忍刹车的响应很慢，那会出事故的。也正是因为实时性的要求，自动系统一般都运行在实时操作系统上，甚至运行在裸设备上。

而且，控制器会跟一个上位机连接，并通过某种工业协议进行通信。上位机的计算能力和存储能力都会比控制器更强。我们通过上位机，就可以了解每个控制器的运行情况，可以

对控制器做一些操作，比如升级控制器上的程序等等。

好了，这就是自控领域最基本的计算架构，以及系统的运行场景。那么为了满足自控系统的需求，需要什么样的开发工具呢？是不是用普通的编程语言和开发工具就能满足自动的需求呢？

自控系统的技术特征

自控程序的开发和运行有特殊的要求。我们平常用到的开发语言和开发工具，很难满足这些需求，因此我们必须设计专门的工具。我来给你分析一下。

首先，自控系统运行的硬件环境是不同的。

从硬件架构来看，这些控制器的设计跟我们平常的手机和桌面电脑都不一样。。

从硬件的电气特性来看，这些硬件在环境适应上的要求要比普通民用的更加严苛。这些硬件在设计上能够更好适应比较恶劣的工业环境，比如能承受比较大的震动、电磁、灰尘等的影响。普通民用的硬件设计，在这种环境下可能会从硬件产生出错误的信号，或者存储的数据发生错误，甚至无法正常工作。

从编程者的角度看，这些控制器的 CPU 的架构也跟我们手机和电脑不大相同。CPU 架构的概念我们前面已经学过了。不同的 CPU 架构，意味着不同的指令集、不同的寄存器、不同的内存寻址方式，以及不同的异常处理机制等等。

不同的 CPU 架构，自然会导致程序开发方式的不同。普通的编译器，有可能都不支持这些芯片，或者没有针对这些硬件进行专门的优化。这个时候，如果你掌握了一些编译技术的后端知识，就知道如何以最优的方式生成机器码，并充分发挥硬件的能力。

第二，自控系统运行的软件环境也是不同的。

我们前面提到过，对于实时性要求比较高的场景，自控程序通常运行在实时操作系统里，或者是在裸设备上运行。所以，我们的开发工具也要能够生成在实时操作系统或裸设备上运行的程序。

不知道你还记不记得，我们前面讲过，在操作系统上运行，需要遵守相应的 ABI，让程序能够跟操作系统相互配合起来。所以，这就需要开发工具支持实时操作系统才可以。当然，如果运行在裸设备上，那么就需要我们的开发平台本身来提供一些运行时功能，有支持 IO、任务调度等能力，那这个运行时在某种意义上也是一个简化版的实时操作系统。

第三，自控系统的软件更新机制是不同的。

自控系统跟我们其他软件一样，在开发过程上也需要进行版本迭代。不过它的版本迭代有时候要求更高。比如，像发电机组等设备一旦投入运行，是不可以随意停机的。每次停机，都会造成很大的成本和影响。还有像卫星、太空舱上面的关键设备，大概也是这样的。这就需要自控系统能够在不重启的情况下实现更新。在这种需求下，自控软件的更新可能会有下面几个特点：

第一个特点：通常这些系统的更新都不是整体更新，而是支持一个个小模块的动态更新。

如果我们拿 Java 语言来打个比方，这相当于程序在运行的时候每次只更新一个类。从这个角度看，我们在课程里讲过的动态编译并形成可运行的模块的技术，就会派上用场了。但是静态编译并不能满足这个场景的要求，因为静态编译后的程序，在代码区里的内容通常是固定的、不能修改的。

第二个特点：在更新的时候，要满足实时性的需求。

在自控系统上，通常会有这样的场景：在 10 毫秒前的一次调用，可能使用的是前一个版本，在下一次调用，就已经无缝切换成了新版本。在这种情况下，程序之间不是静态链接在一起的。当我们要执行某个函数的时候，需要动态查询这个函数的地址，然后再跳转。在采用 JIT 技术的系统中，通常也会采用类似的技术。

第三个特点：在模块更新的时候，要能够保留程序中原来的状态信息。

自控软件在控制设备运行的时候，需要正确地掌握设备过去的状态。拿我们熟悉的场景来举个例子，如果你要升级红绿灯的控制软件，那么程序中的状态信息就要包括每个灯的状态是什么、已经持续了多久。

如果按照原来的规则，红灯要在 5 秒后转绿，那么软件升级后，必须仍然是在 5 秒后转绿，系统整体的运行逻辑才不受影响。否则的话，可能会出现路口两个方向都是绿灯的错误状态，从而引发交通事故。这种状态的错误，如果出现在电梯控制上、发电机组控制上，都会导致致命的后果。

第四、自控系统对可靠性和安全性，有着很高的要求。

通过前面的描述，你应该已经体会到了，自控系统对于可靠性和安全性，有着很高的要求。我们可以想象一下，在未来的战争中，可能我们受到的第一波攻击并不是从天而降的导弹，而是来自网络的攻击。如果网络攻击就瘫痪掉一个国家的高铁、核电、各种基础设施和工厂，那么战争也就根本不用打了。

可靠性和安全性要通过多个方面的工作来保证，包括管理角度、物理防护角度等等。落实到 IT 技术上，也会有多个层面的工作。比如，在操作系统层面，我们会进行可靠性方面的增强，尽量消除由于硬件的原因而产生的数据错误，比如网络传输中的数据错误、由于存储设备的原因导致的数据错误，等等。

从软件开发和运行的角度，在可靠性和安全性方面也有很多可以提升的地方。比如，传统的软件运行方式，代码段的地址都是固定的，所以就会比较容易导致攻击，比如内存溢出攻击。如果你的运行机制，让每个函数的代码地址都是随机的，那么就可能避开这种类型的攻击。

通过我们前面这些对自控程序的运行特征的分析，你会发现它确实跟我们普通的桌面软件、服务端软件、移动 APP 软件都不太相同。所以，针对这个领域，我们就需要专门的开发工具。那我们就看看这个领域的开发工具都有哪些特点。

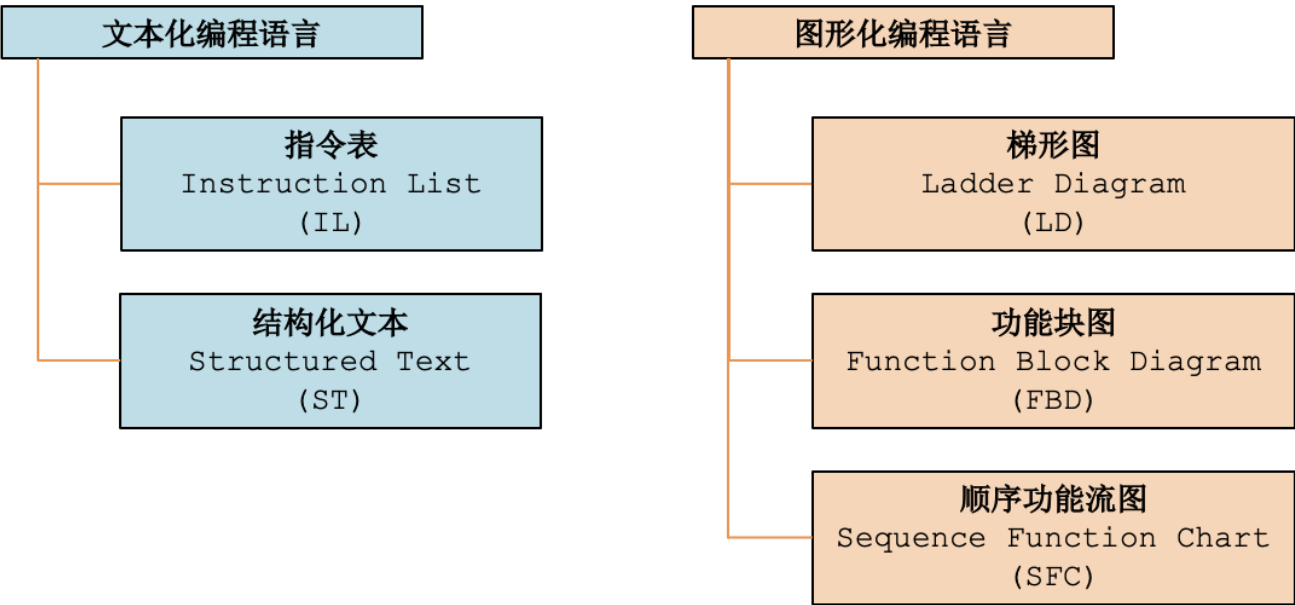
自控系统的开发平台

由于自控领域自身的独特性，所以多年来这个领域也发展出了一系列独特的技术。这些技术被统称为 OT，也就是 Operational Technology。而你熟悉的互联网系统等，则属于 IT 领域。在 OT 领域，也形成了相应的国际组织和技术标准。在过去，这些组合和标准的话语权主要在国外一些企业的手中。

在 OT 领域，软件开发被叫做“组态”开发。组态是英文 Configuration 的意思。从字面上你可以这样理解，我们给控制器开发的软件，相当于是对控制器在做配置。

而这些控制器呢，在 OT 领域有个广为人知的名称，叫做 PLC，也就是可编程控制器（Program Logic Controller）。那么我们可以怎么为可编程控制器编程呢？

根据相关国际标准（IEC 61131-3），如果我们要给 PLC 编程，可以使用 5 中不同的编程语言，如下图：

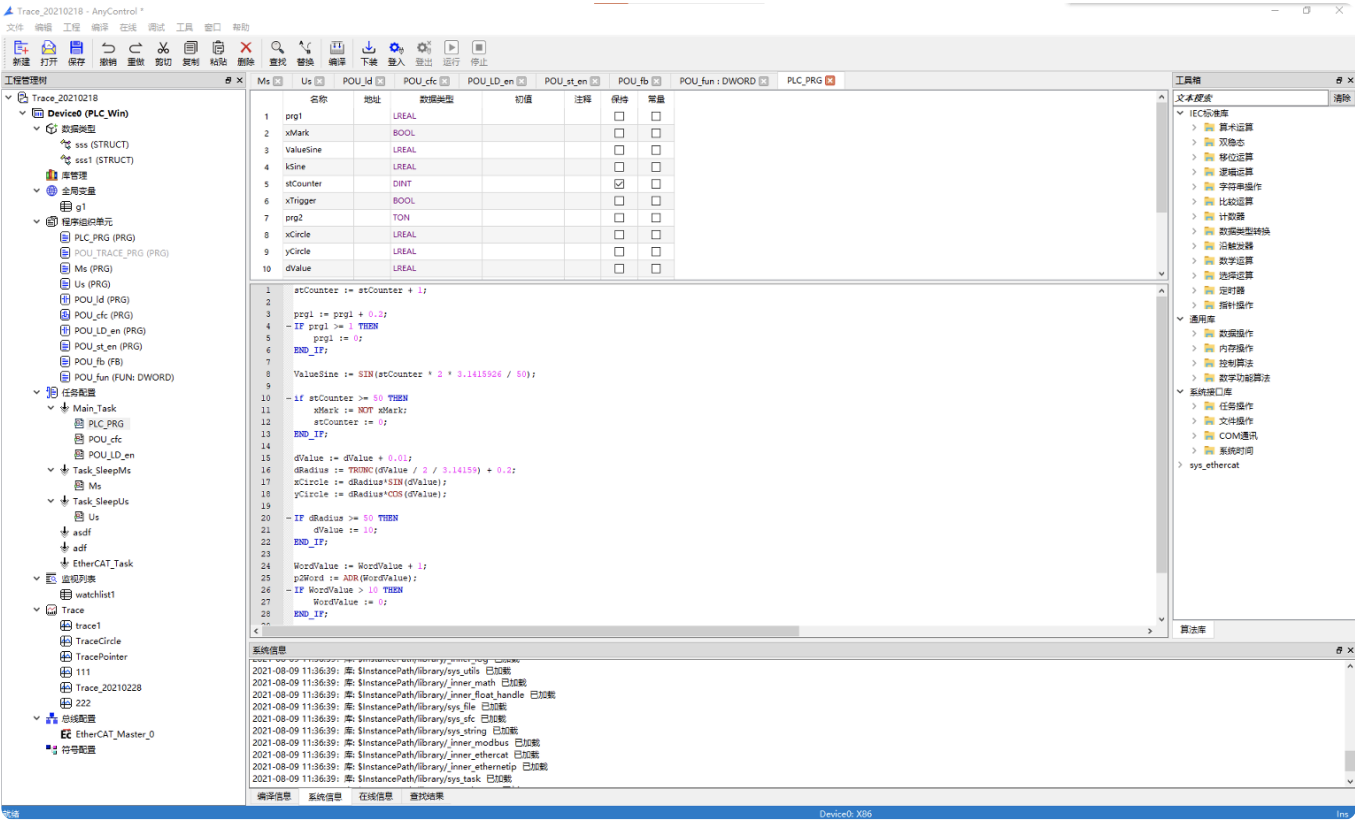
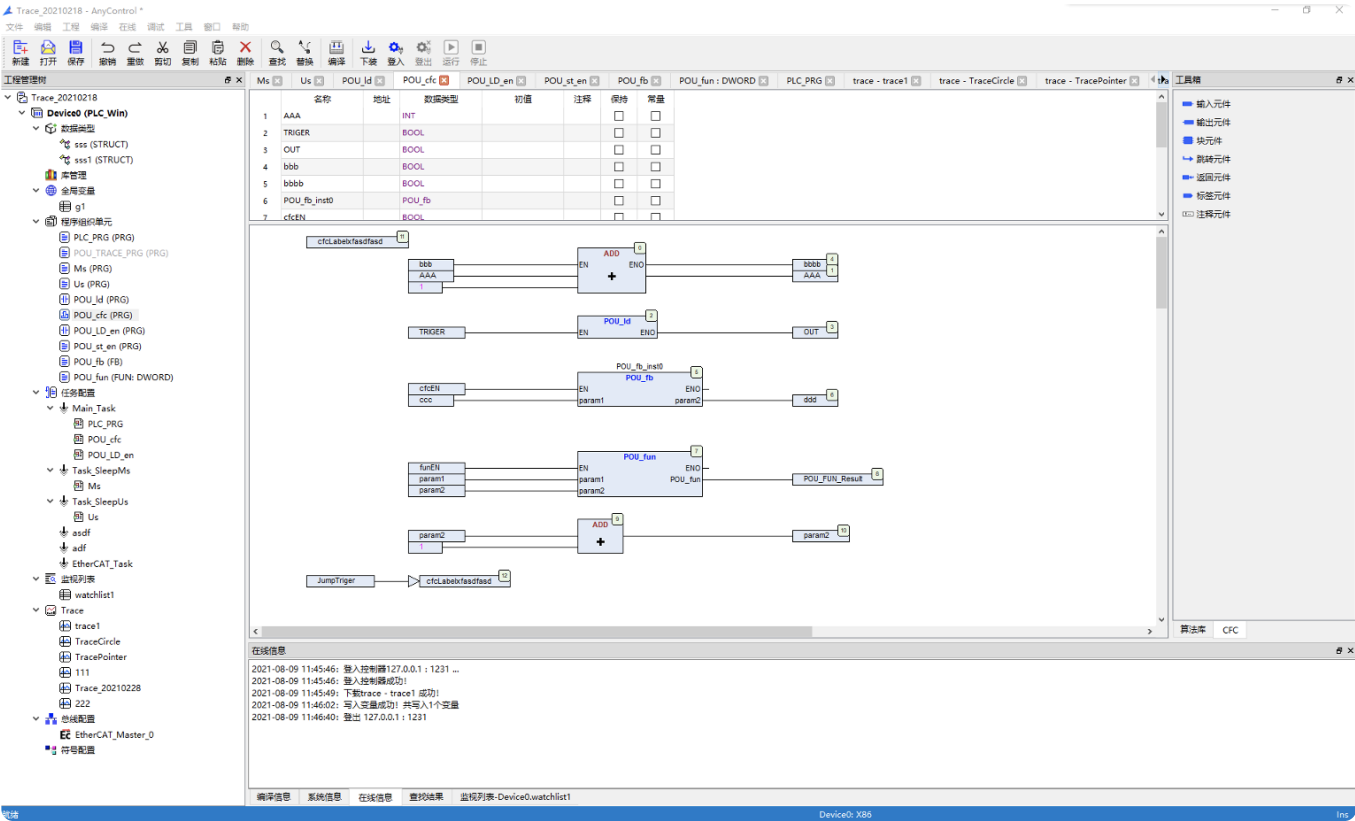


首先是文本化的编程语言。我们平常用的编程语言其实都是文本化的，这里面又细分为两种。一种是指令表，它相当于一种语法特别简单的计算机语言。虽然简单，但是对某些编程需求来说却足够。第二种是结构化的文本，这个跟普通的高级语言差不多，语法上有点像 Pascal 语言。

除了这两种文本化的编程语言以外，还有**三种图形化编程语言，包括梯形图、功能块图和顺序功能流图。**我相信，这几个图对大多数 IT 领域的同学来说，应该都不太熟悉。但对于弱电领域或 OT 领域的很多工程师来说，阅读这些图是他们的基本功。

通过这些图，他们就能很容易地理解程序的逻辑。其实在 IT 领域，一直也有图形化编程的方式，比如少儿编程领域，就可以通过拖拉图形块的方式来编程。而最近越来越为人所知的低代码开发平台，也大量采用了图形化编程的方式。

我这里给出了几张图，是对某个组态开发平台的截屏，可以帮助你更直观地理解这些编程语言。



那么，问题来了，你可以怎么来利用我们这门课学到的知识点，来实现上面这样一个开发平台呢？

首先，针对这 5 种语言，无论是文本化的语言，还是图形化的语言，你都可以利用词法分析、语法分析和语义分析技术，形成统一的、正确的 AST。

第二，基于这个统一的 AST，你再可以继续编译成机器码。我们在课程里讲过了如何针对 x86-64 架构的 CPU 生成汇编代码。不过呢，你要根据 PLC 所采用的芯片，来为它生成针对该芯片的汇编代码。一般芯片厂商都要提供工具链，能够把相应的汇编代码生成机器码。

第三，我们要设计一个专门的运行时。在我们的课程中，我们已经实现过虚拟机，这就是一种运行时。不过，针对 OT 的需求，PLC 的运行时要复杂一些，比如要能够实现软件模块的动态加载和更新的管理、代码地址的转换、状态信息的维护，还有与通讯模块和底层操作系统的衔接等工作。

第四，我们要对程序进行优化。对于比较复杂的控制逻辑，我们要运用优化算法，提高程序的性能。程序的性能越高，可以满足的实时性要求就越高。像控制机器人臂这样的场景，对实时性要求就是很高的。如果我们不把代码进行充分优化，那就很难满足这些高实时性场景的要求。

所以，我们在课程的第三部分，会专门花时间学习优化技术。如果你能把优化技术也掌握透彻，那么你就有可能成为这个领域的顶级技术专家了。

通过上面的这些分析，你会发现，其实很多知识点，我们在前面的课程中都已经涉及了，还有一些知识点我们会在后面继续学习。所以，只要你认真掌握了我们这门课的内容，你基本上就可以胜任这些技术工作了！

课程小结

今天的加餐，我分享了自动控制领域的一些背景信息，也讨论了如何针对这个领域的需求来研发相应的组态软件平台。我写这篇加餐其实有几个目的：

第一，是开阔你的视野。你可能并不是 OT 领域的技术人员，以后也不会做 OT 有关的事情。但是，它山之石可以攻玉。OT 和 IT 在发展过程中，一直在互相影响。比如，OT 处理高可靠性的一些思路，就有可能用于高可靠的 IT 应用中。

第二，我希望你能理解我们这节课的知识点，是怎么用于解决具体领域的问题的。比如，你可能发现，我在这门课里特别重视让你理解程序的运行机制，包括程序跟 CPU 架构的关系、跟操作系统的关系、理解 ABI 等等。从今天的分享中你会看到，要解决一个特定领域的问题，特别是当你需要自己研发相关工具的时候，这些知识都很重要。

最后，我也希望通过今天这节课，能让工业领域之外的人也了解一点工业软件。毕竟，工业是我们国家的立身之本。而工业领域的基础软件，还有很多工作，需要有志之士参与进来，提升我们国家在这个领域的创新能力和话语权。

我在后面的加餐里，还准备给你分析一些其他的应用领域和开发工具。我们总说“学以致用”，了解更多的应用场景，也会有助于你理解和掌握我们这门课程的知识点。

思考题

如果你是来自于自控领域的，我想请你帮我补充一些信息，包括自控领域的其他应用场景、编程技术等等。

如果你来自其他领域，那么我想问问，你们的领域有没有跟自控领域类似的技术问题，以至于需要研发专门的开发平台呢？欢迎在留言区和我分享。

我是宫文学，我们下节课见。

分享给需要的人，Ta订阅后你可得 **20** 元现金奖励

 生成海报并分享

 赞 0

 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 [“它山之石” | Sugar：这门课你可以试试这么学](#)

精选留言

 写留言

由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。