

人物专访

专访C++发明人Bjarne Stroustrup*

关键词：程序设计语言 C++

金芝 刘譞哲
北京大学

本贾尼·斯特劳斯特卢普 (Bjarne Stroustrup) 博士是 C++ 语言的设计者和最初的实现者。C++ 自 1978 年问世以来已成为使用非常广泛、影响力

颇大的面向对象程序设计的语言。本贾尼博士所著的《C++ 程序设计语言》广受欢迎，已被翻译为至少 19 种语言，是国内外很多重点大学的 C++



* 由于篇幅所限，本文所登载内容为编辑过的版本，略去了一些技术细节，Bjame博士访谈的英文全文请参见 <http://www.stroustrup.com/interviews.html>。

语言教材。他的代表作还包括《C++ 程序设计原理与实践》、《C++ 语言的设计和演化》、*A Tour of C++* 等。

本贾尼博士出生于丹麦奥胡斯，目前定居于美国纽约。他于1975年获得丹麦奥胡斯大学数学与计算机专业硕士学位，1979年获得英国剑桥大学计算机专业博士学位，并于同年加入美国新泽西贝尔实验室，截止到2002年，他一直担任贝尔实验室大型编程语言研究部的负责人。2002年至2014年，本贾尼博士受聘为德州农工大学工程学院计算机科学首席教授和哥伦比亚大学计算机科学系客座教授。

多年来，他为剑桥、哥伦比亚和普林斯顿等知名学府的计算机科学教学事业倾注了大量的热情和精力。本贾尼博士在工业界也很有名望，是RollApp科技公司的资深顾问。2014年，本贾尼博士加入摩根士丹利(Morgan Stanley)公司，在纽约总部担任技术与数据部门的董事总经理。

2014年9月，本贾尼博士访问了北京大学软件研究所，与北大师进行了座谈，分享了他关于C++语言设计的宝贵经验和在研究教育实践中的心得体会。为此，我们也有机会对他进行了专访。

关于程序语言设计

问：您最初设计C++的目标是什么？在设计C++的过程中，让您记忆最深刻的是什么？

答：C++最初（事实上现在也是）的设计目标是让代码执行更加高效，合理地控制系统复杂度，以及提供直接操作硬件而不引入额外开销的抽象能力。如果让我指出C++的一种最重要的特性，那一定是构造函数和析构函数，它们是C++语言大多数特性（如隐式和可预测的通用资源管理、C++的异常安全性等）的主要支持机制。但有意思的是，构造函数和析构函数的设计并不困难，它在C++设计的早期阶段（在1979年或1980年）就已经是一个著名的语言特性。

问：您如何评估一种程序设计语言，换句话说，是什么造就了一门“优美”的程序设计语言？

答：许多年来，我的目标一直是设计一门“有用”的语言，而不仅仅是一门“优美”的语言。当然，如果可能的话，我也欢迎“优美”的程序设计语言，但不能以牺牲性能、兼容性和通用性等为代价。我认为，“优美”的度量方式是在多大程度上告诉程序员应该做什么，而不是应该怎么做。事实上，如果不考虑通用性和性能的需求，“优美”的语言并不难实现。

问：作为一门编程语言的设计者，无疑您对计算机系统底层的细节是非常了解的。那么，您如何看待形式化方法？形式化方法又如何能够应用于C++语言的设计与实现？

答：多年以来，我和我的朋友们曾多次尝试使用形式化方法去完成一些任务，如定义内存模型、常量表达式的赋值、定义模板参数需求（如Concepts）等。但是，并没有一个形式化方法或者模型能够应用于整个C++语言，其主要原因在于C++的抽象是基于定义同物理硬件的接口，而非自底向上的数学抽象。C++的确缺乏一个介于程序员代码和物理硬件的中间抽象层次，这是C++的不足。

问：在过去很多年里，学术界做了大量的努力，从形式化方法、程序设计语言、运行时环境以及软件工程等多方面研究可信计算。从程序设计语言的角度，您觉得我们应该如何增强软件的可信性？

答：首先我们必须改变如下观念：程序设计是一种低层次的工作，最好由成本低廉的劳动力来执行。但在实际工作中，如果没有严谨、专业、高效的程序设计，开发者往往会采用一些权宜的解决方案来控制成本，而不是将软件质量作为首要考虑，这是一个很严重的问题。

我认为，我们要在完整的类型安全性和资源安全性上下功夫。我们要使用更强类型的代码风格，这一

点非常重要。例如，由于一个简单的计量单位上的失误¹，我们损失了价值6亿5千万美元的火星气候探测器号卫星。此外，我们需要更好地处理并发问题，在这一方面，静态分析是很有意义的研究领域，尤其是将静态分析和代码生成结合起来的研究。

需要说明的是，我个人不推崇以增强安全性的名义在语言中加入过多的限制，或者引入昂贵的运行时开销。事实上，我们正努力在C++中引入更好的和具备更强表述能力的类型系统，并且努力不增加运行时的额外开销。

特别值得一提的是，我们必须记住，安全性、可信性、可靠性是整个系统的属性，其实现不能仅仅依靠程序设计语言。

对程序设计语言未来发展的展望

问：在编程语言方面，您认为最有前途的方向是什么？或者您认为最有趣或最重要的问题是什么？

答：我们必须解决并发编程的各种问题，特别是要提供比已有的线程和锁机制更加简单、高效的编程模型。ISO标准委员会中的一些研究小组正在做这方面的研究。比如，在C++11中，我们支持无锁编程以及类型安全的线程和锁。我现在可以提前透露一点，在C++17中，将会有一些非常重要的新特性，例如支持工作窃取(work-stealing)、协程(co-routine)、单指令多数据流(SIMD)、事务性存储(transactional memory)的程序库。

从编程语言的角度来看，我们还需要在不降低C++语言能够直接处理硬件能力的前提下，增加类型安全性以及可表述性。显然，性能也需要提升，也就是应该大幅缩短编译时间。ISO标准委员会中的一个研究小组正在进行这方面的研究。

问：提及并行，事实上我们的硬件已进入多核

时代，但是程序设计语言对并行的支持仍不能令人满意。您觉得如何增强现有的程序设计语言以支持并行？下一代程序设计语言和语言的设计者应坚持哪些基本的原则？

答：并发（通常是多核的形式）和并行（通常是增强CPU中浮点计算的形式）是现代处理器设计中性能优化的主要方向。新的程序设计语言也应该在这方面有所进展。

我认为程序设计语言必须增强对并行和并发的支持，这一点毋庸置疑。今天，我们在最底层的通用层次（线程和锁）实现并发的支持。但是我们可以做得更好。我们不应该将高层次的并发模型设计成语言的一部分。相反，我们应该在程序设计语言层次支持相对底层的并发机制，并在此基础上设计高层次的类库。C++11提供了可迁移的无锁编程的原语、类型安全线程和锁的支持。我期望C++17会提供若干高层次并发和并行的模型。

问：面向对象程序设计语言仍然是信息技术领域最重要的语言。最近我们看到工业界开始采用函数式语言和动态语言，如Scala和Ruby。您对未来几十年工业界使用的程序设计语言有何看法？

答：我不认为一种程序设计语言或者一种编程范型能够解决所有问题。现实世界有很多具有挑战性的问题，最好采用多种程序设计语言和编程范型来解决。

事实上，我在设计C++的过程中，就采纳了多种语言中若干有效的技术，并尽量将这些技术融会贯通。我写的许多程序中也混合采用了面向对象和面向方面(aspect)的程序设计、函数式程序设计，甚至基于过程的程序设计。

我希望工业界和学术界意识到，不同的开发任务需要的不仅是不同的程序设计语言，而且需要不同的能力和不同的教育。

¹ 火星气候探测器号任务失败的主要原因是人为因素，因为火星气候探测器号上的飞行系统软件使用公制单位“牛顿”计算推进器动力，而地面人员输入的方向校正量和推进器参数则使用英制单位“磅力”，导致探测器进入大气层的高度有误，最终瓦解碎裂。

问：对于设计新的程序设计语言，您认为最重要的原则是什么？

答：确定新的语言所要解决的问题，并确保你做得足够好。此外，我更喜欢具有类型和资源安全性、表达能力更强、对互操作性支持更好的语言。

仅仅为了对现有语言进行较小的改进而设计和实现一门新的语言基本上是浪费时间，并且可能对已有系统有破坏性，往往很难成功。所以，一门新的语言需要解决一个新的问题，或者相对于已有的语言，能够提供一种显著的更好的解决方案。

但也永远不要忘记，如果一门新的语言看上去有用，老的语言也会采纳新的语言特性或者类库。

问：您认为我们如何通过众包 (crowd sourcing) 来增强现有的程序设计语言？

答：众包适用于已充分理解的问题以及成熟的工具和过程，在很多领域中取得了成功。但对程序设计语言的设计本身而言，众包可能不是一个很好的方式。“群众”对新事物有无限的好奇心，但往往缺乏足够的鉴赏力，并且不太理解程序设计语言应该长期稳定的需求。但是，作为程序语言设计者，我们确实可以通过众包受益，比如从更多人实现的应用或测试用例中不断改进语言本身。

给研究者的建议

问：对于从事程序设计语言、形式化方法和软件工程的博士研究生，您有什么建议？

答：在我当初计划选择上述领域作为研究方向时，莫里斯·威尔克斯 (Maurice Wilkes)² 曾给我很中肯的建议。我们应该着眼于解决一些具体的应用问题，如网络、信息安全或者图形学中的具体问题，从这些具体问题着手研究如何对程序设计语言和工具进行改进。我个人认为，在程序设计语言领域，大多数刚毕业的博士研究生并没有足够的经验来判

断他们取得的进展是否真正有用。他们急需工业界的经验和领域知识。这应该引起这些领域研究人员的足够重视。

问：最后，您还想和中国计算机科学领域年轻的研究人员分享什么？

答：作为计算机科学研究者，你应该关心如何去改变世界，而不是只关注起薪和职位。在工业界和学术界工作了这么多年之后，我仍是一个理想主义者。你可以认为我很天真，但是这个世界需要天真的人。重大的改变需要时间，往往需要几十年。找到一个值得投入的“伟大目标”，并在接下来的时间持续专注下去。职业发展和家庭的需要很有可能会不时打断这个目标，但是永远不要忘记你的理想。脚踏实地，胸怀梦想，这对每个人来说都是重要的。

我们的文明依赖于计算机及其软件和硬件。软件要可靠、可信且价格合理，否则我们都会有麻烦。

另外，不要只专注计算科学领域，其他领域可以激发你在计算领域的灵感。你希望计算机如何做得更好？如何才能让世界变得更好？可能你需要学习史学、生物学、社会学、数学、音乐等。不要在计算的技术细节中迷失。■



金 芝

CCF常务理事、会士。北京大学教授。主要研究方向为基于知识的软件工程、形式化方法、高级程序设计语言等。
zhijin@pku.edu.cn



刘 贇 哲

CCF会员。北京大学副教授。主要研究方向为软件工程、服务计算、Web程序设计等。
liuxuanzhe@pku.edu.cn

² 莫里斯·威尔克斯：英国计算机科学家，世界上第一台存储程序式电子计算机EDSAC的设计者，1967年ACM图灵奖获得者。