

个人资料



zjufirefly

访问：6426次

积分：464

等级：BLOG 2

排名：千里之外

原创：42篇

转载：4篇

译文：0篇

评论：1条

文章搜索

文章分类

C++ (14)

其他 (5)

链接 (13)

linux (13)

shell (7)

操作 (11)

文章存档

2014年12月 (8)

2014年11月 (7)

2014年10月 (8)

2014年09月 (6)

2014年08月 (4)

展开

阅读排行

git操作命令 (327)

blktrace (287)

mysql常用命令 (280)

python (270)

cgroups (262)

linux下如何配置openvpn (260)

2015年4月微软MVP申请

《京东技术解密》有奖试读，礼品大放送

"我的2014"年度征文活动火爆开启

CSDN 2014博客之星

《C++ Primer》作者Stanley B.Lippman谈C++语言和软件产业的发展

分类：C++2014-05-02 11:4895人阅读评论(0)收藏编辑删除

目录(?) [+]

C++ Primer》作者Stanley B.Lippman谈C++语言和软件产业的发展

发表于 2013-12-04 11:06 | 31580次阅读 | 来源 《程序员》 | 119 条评论 | 作者 高博，范德成

《程序员》杂志2013年12月刊c++计算机科学软件工程师Stanley B.Lippman

摘要：《C++ Primer》作者Stanley B.Lippman到访中国，并于10月27在由CSDN主办、电子工业出版社博文视点协办的CSDN TUP大师系列线下活动上做了主题演讲，介绍了C++的新进展，本刊特约记者对Lippman进行了采访。

金秋10月，《C++ Primer》作者Stanley B.Lippman到访中国，并于10月27在由CSDN主办、电子工业出版社博文视点协办的CSDN TUP大师系列线下活动上做了主题演讲，介绍了C++的新进展。活动间隙，本刊特约记者对Lippman进行了采访，形成此文。



勉励中国程序员要加强创新

《程序员》：您好，Lippman先生，欢迎来到中国。我们上次见面，还是在2004年的微软TechEd期间，如今已经过去近10年了。您感觉中国软件行业 and 软件社区有哪些变化？

Lippman：事实上，我2009年还来过一次中国，那时在上海首次发表了现在称为Hugo的研究项目的主题演讲。

这些年，我所认识的大多数曾在微软等美国企业工作的中国人，现在都工作于中国国内企业了。而中国开发者社

blog.csdn.net/zjufirefly/article/details/24870761

1/6

vim-ctags-taglist-netrw	(249)
性能测试命令字段解释	(213)
正则表达式	(211)
SystemTap	(195)

评论排行	
为什么需要auto_ptr_ref	(1)
valgrind内存检查	(0)
cgroups	(0)
linux下如何配置openvpn	(0)
git操作命令	(0)
thrift	(0)
Apache开源软件	(0)
vim-ctags-taglist-netrw	(0)
vim常用配置	(0)
shell按行读取字符串，并	(0)

推荐文章	
* 挣扎与彷徨——我的2014	
* 校招回忆录——小米篇	
* Android UI-自定义日历控件	
* 30岁程序员回顾人生、展望未来	
* 2014年终总结，我决定要实现的三个目标	
* Android 启动问题——黑屏 死机解决方法	

最新评论	
为什么需要auto_ptr_ref	
zjufirefly: explicit auto_ptr_ref(Tp1* __p): _M_ptr(__p){}...	

区则明显表现出了更强的自信心，或者说独立性。我是说，在2004年时显得更偏向于由西方主导，而现在更偏向于中国国内主导。

《程序员》：那么，您认为这种自信心源自哪里？

Lippman：源自他们取得的成就。例如，你们已经有了自己的如同Facebook一般的社交网络工具，也有了自己的互联网。我2009年就去过中国的一家大型公司，深刻感受到它做得非常成功。在那家公司里，我遇到不少曾在微软研究院的熟人，他们已经加入了那家公司了，而且他们显得信心满满，觉得自己就代表着未来。

《程序员》：您是否认为，现在中国的软件社区和是否越来越不受西方驱动了，并且拥有它们自己的想法和创新意识了？

Lippman：我认为这还有待见证。这就好比自由软件基金会。光有设想还不能算是创新。自由软件基金会其实没做过太多有意思的事情，他们所做的大多数工作就是克隆现有的商业软件。而从目前看来，中国的软件社区做出来的有意思的事情也比较有限，它们也还是在创建一整套业界现有软件的仿制品。

但我期望看到的是，你们能够做一些创新的事情。目前而言，大多数的软件创新都还来自别处。你们要看一下自己的特色是什么。美国计算机行业现在变得相当软弱，人们不再勤奋地工作。他们被惯坏了，只想快速赚钱，并且不再努力工作。中国的优势，或者说中国程序员的优势在于，他们很勤奋地工作，并且渴望成功，愿意遵守纪律，也很投入工作。而在美国我看不到这种气氛了。

所以我认为，你们有着足够的能力在你们的国家做出伟大的事情，但这个时刻还不能说已经到来了。

《程序员》：我觉得您的说法是要鞭策我们不只是勤奋地工作，还要聪明地工作。

Lippman：就是这样。

回忆C++语言诞生时期的峥嵘岁月，在贝尔实验室的黄金年代

《程序员》：您刚才说到，美国业界的目前情况是，人们不再努力工作。但我想，在30年前肯定不是这样。而您则经历过那个时代。我不知道您是否还记得，C++编程语言最初的那些实现中，是否有一些完全由您实现出来的功能，如果有的话，您能否告诉我们您是如何设计它，以及它的哪些部分比较有意思呢？

Lippman：我一向热衷于编译技术。当我开始与Bjarne合作工作于Cfront时，我在1983年从Jerry Schwartz那里了解到了C++。当时Jerry工作于Steve Johnson的一个名叫NAIL的项目，即“新一代中间语言”，全部由C++实现，而我当时则在从事C++语言组的测试工作。然后，在1985年，Bjarne决定发布Cfront，即后来被称为“发行版1.0”的实现。之所以这么急迫是因为，如果C++老是处于实验阶段，而没有官方技术支持的话，那么大学里的研究生就无法使用这门语言来做研究。我们为了让C++能够成功发布而很努力地工作着，而我做的是测试工作。我问了一些很有难度的问题，他对我做测试时的思路感到满意。

《程序员》：“他”是指Bjarne还是指Schwartz？

Lippman：Bjarne，他因为Cfront发行版1.0的工作而注意到了我，因为我总是询问有关该语言的问题。你知道，那时我们对C++做了种种严苛的测试来发现Bug。而我还 会问他一些微妙的问题，诸如某种特定代码的行为应该是怎样的，所以当我从测试小组转到编译器小组时，我可以选择做代码生成——这正是我感兴趣的——或者做Cfront。我当时不太想做Cfront，因为我对于前端没有兴趣，因为前端的逻辑很混乱，我是说，它没有一定的规则。而后端的工作，则有逻辑、有算法得多。

我的朋友Barbara E. Moo当时是我的导师，我早已认识她多年，是她说服我接手Cfront。实际上Steve Dewhurst也是我的好朋友，我知道你（高博）翻译过他的一本书《C++语言：99个常见编程错误》。我在贝尔实验室的两位最要好的朋友就是Barbara E. Moo和Steve Dewhurst。Steve曾作为实习生与Bjarne一起工作过，因为他曾在贝尔实验室从事C++编译器方面的工作。

当时他和Bjarne坐在同一间办公室里，办公室的大小与一般的宾馆房间差不多。他们并不总是相处得很好。当我开始做Cfront上的工作的时候，已经是1986年的1.1版的事情了。那时，有一个名叫BlockingsField的以色列公司，他们的一些员工当时正把Cfront移植到PC平台上，那个年代的PC使用的是16位段式体系结构，这是因为地址只有16位宽的缘故，而在Unix上我们使用的是32位地址。我们都不希望与PC打交道，因为都觉得它是垃圾——从计算机科学的角度来看的话。但我们别无选择，必须发布一个PC版本，于是我们只得做一些事情来使它变为可能，比如，添加了无符号整数。我是说，在1.1版发布之后我们又不得不添加了一些东西。

《程序员》：就是说，C++中的无符号整数是您添加的？

Lippman: Cfront 的最初版本中是没有无符号整数的，Cfront是Bjarne的一个研究项目，所以它当时并没有像一个正式发布版的那种产品级成熟度。例如，如果你使用 Cfront，而我们还没有实现某种特性，我们可以让它显示“对不起，还没有实现”或者我们会说“如果你这样做的话，会有小鬼缠身哟”。你知道，我们在正式发布版里面不可能这么干，但我们一开始做的并不是正式发布版。因为最初的Cfront只要50美元就能获得源代码，几乎就相当于免费了，所以任何人都能 购买C++完整的源代码。

Bjarne知道，我可以与同事们和睦相处，因为我当时比较年轻，而我又不懂计算方面的东西，所以我从不与他争论，因为他懂的东西是我所知的上百倍！在1.1版以后，他作为研究者，继续他有关多重继承特性的工作，而我是唯一的Cfront支持人员，也是唯一的开发人员。所以基本上大家所知的关于C++的每个特性：成员函数、指向成员函数的指针等等，都是我实际地把它们一个个添加到Cfront中去的。所以那是一段 对我而言非常激动人心的经历。我和Bjarne相处得非常融洽，因为我不喜欢问问题——我喜欢自己研究出问题的答案，而他也不太喜欢回答问题。所以他不管我，我也不打扰他，大家反而相处得很好。

C++语言从实验室步入成熟所经历的阵痛

《程序员》：您是说，在早期的C++语言实现中，像指向成员函数的指针这样的特性，都不需要通过一个非常正式的评审过程就能加入吗？

Lippman: 当时完全不具正式性。拿Cfront发行版1.0来说，我测试了它，写了所有的文档，参与了它的移植。所以当时我像是Cfront的单人乐队，因为在产品 部门内部没有任何支持，也还没有人需要这种支持。当时所有人都说：我要用Fortran 77，或者我要用Ada，没有人关心C++。

Steve Johnson被调离以后，新上任的部门经理Carlson Rillo完全不了解计算机语言，所以他所知道的所有事情就是，他需要Ada以获得认证，从而获得政府部门的合同；他需要Fortran 77，因为有一些公司想要买。但他不知道，也不关心C++。事实上，他就是当初那个问我老板Barbara奇怪问题的家伙：“为什么Lippman在写一本关于C++的书？他到底是什么来头？”

我们都看不起这家伙，他是个打高尔夫的——此人不过是被强行委派到我们小组头的人物，并无远见。只是因为Steve当初管的事儿范围太大了一点，所以才失控了，但在我眼中，Steve完全称得上是一个研究员，而Carlson Rillo则是一个彻头彻尾的职业经理人。不管怎样，总地来说那段时间是非常精彩的，是我职业生涯中最美好的日子。

《程序员》：随后在迪斯尼和微软的经历如何？

Lippman: 我在迪斯尼工作的那段时间可就没那么美好，你也看得出来，我在一个高度组织化的环境中工作得并不顺心，那里有很多条条框框。所以，当我一开始工作于Cfront上时，那里并没有规则，只有我和Bjarne。

刚来到迪斯尼时，我是首席软件工程师，还有一位是Peter De Vroede，我们一起参与软件工程，并且是那里仅有的来自计算机科学背景的软件工程师，而其他每个人都是动画师、动画制作人员，像Scott Johnson、M. J. Turner等等。他们都是非常聪明的人，但他们不是工程师，所以他们写出来的代码很不修边幅，而我在那种环境中能发挥出我的能力。但随着环境变得越来越结构化，就很难再有机会做些什么事情出来。当所有东西都变成了委员会机制时，我在那样的环境中就不能出色地工作了。

这也就是为什么我在微软干得并不出色，因为在微软，程序员不能决定任何东西，能作决定的是项目经理（Program Manager），他们是接触用户的那些人。但用户并不知道他们想要什么，他们只知道他们现在有什么、现在喜欢什么。例如，对于Cfront，我可能接到一个开发人员打来的电话，他说他遇到了一个Bug，我就会修复它，并发送一个补丁给他。我不需要通过什么审批，他们马上就能得到这个补丁。但在微软，可能需要等上两年才能得到一个补丁。因为它必须通过委员会的审批，所以必须等待相当长的时间才能被发布。并且Oracle这样的企业用户，虽然需要等待，但你还是能得到一个补丁。可是如果你只是一个大众开发者，你将被彻底无视。所以我对你问题的回答就是，我们一路过来基本上都是自己做的，直到标准委员会成立为止。

标准委员会被发展起来以后，生活就不再那么愉快了。例如，这边我和Bjarne想做一个特性，我就会试着实现它，看看它工作起来效果如何。但下一次会议时，标准委员会说，不，我们要收回这个特性。于是不那么愉快了。

《程序员》：确实是这样，当组织越来越大，用户群体爆炸式增长以后，组织里面就有了越来越多的决策者。

Lippman: 这就变得相当无趣。例如，在微软，所有的东西都必须达成共识。这样做的问题在于，微软的很多

人带着他们的笔记本电脑参加会议，他们在会上首次听说某个设想后就发表观点，而他们之前都从来没有听过这个设想。但你知道，对于计算机问题，你必须思考得足够深入，因为它们总会产生种种后果，所以这样不经过深思熟虑是非常令人沮丧的。

我当初的职责是尝试找到把C++移植到.NET上去的方法。我去洛杉矶，在这个问题上工作了个月。然后我向他们提出并演示给他们看。接着，有人会说：“不，这样不好”，但他们之前连听都没听到过这些思想。还有人会说“那样做的用处是什么啊？”所以你问我之前负责哪方面工作，我的回答是负责所有方面的工作：Bjarne发明了那些特性，我负责实现它们。

而有意思的是，因为你实现了它们，而当时又没有任何标准，就很少有人了解那些特性是什么，除非通过编译器来检验。所以，大概四年来，我自己实现GCC的Michael Tiemann、实现Oracle C++编译器的Michael Bob、实现PC上编译器的Walter Bright，我们是当时仅有的几个真正理解这门语言的人，当然，还包括Bjarne。你知道，这也就是为什么人们想要建立语言标准的原因。

但问题是，一旦你有了标准以后，所有事情都变得困难起来。因为，比如说，微软想要异常处理的另一种模型，然后Bjarne觉得技术上这是正确的。他们想要一种恢复语义的执行期模型，这是出于他们自己的考虑，于是事情就变得政治化了。然后整个过程就变成了如何获得你的支持者联盟了，而且，突然间，Bjarne无法思考并做出决策了——他必须在利益各不相同的多个集团之间取得共识，比如，公司有自己的利益，使得我们无法在二进制兼容性上达成一致。

再例如，微软有一个虚拟基类指针实现，实在拙劣得很。但微软为此申请了专利，于是造成巨头之间爆发了标准战争，兼容性就被抛在一边了，这也是为什么微软要发明COM来解决这个问题。因为如果你无法在编译器之间形成二进制兼容性，就意味着人们就必须用PC上的各个编译器都发布一个库，而这几乎是不可能的。

这个工作量太大了，对吧？如果你想要发布一个字符串库，你必须在Borland上实现一遍，又在Zortec上实现一遍，又在微软编译器上实现一遍。然后你又必须在Sun上面实现一遍……因此，在我看来，它已经破坏了C++本身所拥有的优秀特性。所以它给我留下了糟糕的印象。

《程序员》：这听起来与C++设计哲学背道而驰。

Lippman：是啊，但为了成为一个业界的标杆，标准还是需要的，只是这方面的工作已经不再有趣了。这也就是为什么当有机会和Bjarne一起去研究部门做GRAIL项目时，我就抓住这个机会调过去了，因为这是我们一拨人能令兴奋的事情的又一次机会。做那个项目的时候就是《深度探索C++对象模型》一书问世的时候——我当时的工作就是建立那个模型。

《程序员》：如果我没有记错的话，最后Bjarne决定不采纳恢复语义，而转向了中止语义？

Lippman：但这是因为他没有研究过现有的模型。他咨询了好多实现者，求证他们的意见，他已经尽力而为了。



谈天才与后天努力，Lippman眼中最好的C++编译器

《程序员》：您在中国有很多粉丝，包括我在内，都认为您是一个真正的天才。



Lippman: 我不是天才。我很幸运, 而且我工作得很勤奋。但我不是一个天才。

《程序员》: 那么我们能不能说, 你在正确的方向上作出了努力?

Lippman: 有一本书叫做《权力的游戏》, 它也被改编成了一个HBO电视连续剧, 或者说是魔幻剧。里面有一个侏儒角色名叫Tyrion, 他总是不停地读书。他的哥哥是一个勇士, 一个帅哥; 而他的爸爸也是一个勇士, 有权有势。所以有人问他为什么读了那么多书。他回答说: “孩子, 因为我是一个侏儒。如果我没有智慧, 那就没有人会尊重我了。”所以我努力工作因为我必须这样做, 而不是别的原因。因为我来自蓝领阶层, 来自主流文化之外, 没有读过好的学校。我的父母都是外来移民。除了获取成就, 我别无他途。

《程序员》: 您曾参与第一个C++编译器Cfront的开发, 但您对它的实现并不满意, 能谈谈其中的原因吗?

Lippman: 我想实现一个特性, 他们答应我让我来实现异常处理。但组织里的某些人不认为我有能力完成它。所以他们把这项任务转交给惠普。惠普后来交付给我们一个Cfront版本, 在这个版本上, 一个带异常处理的Hello World花了15分钟才编译完成。

《程序员》: 听起来很不可思议, 这样的产品性能是无法接受的。

Lippman: 很遗憾, Cfront以这种方式终结了。如果我来做的话, 无论如何不可能做得比这还糟糕, 对吧? 而这也导致了后来EDG编译器的出现, 因为John Spicer和我曾在同一家公司工作, 而当Cfront失败以后, 他加入了EDG, 他们的编译器就成了业界标杆。它是当时最优秀的C++编译器。而现在最优秀的则是LLVM, 它是一个出自大学的编译器, Google和苹果都对它作了不少投资, 但它仍是一个出色的编译器。它也支持Objective-C, 和Xcode——苹果的编译系统。我认为它是市面上最出色的编译系统。

《程序员》: 原来如此, 也许使用LLVM来测试一下C++11新标准会有价值的尝试。您对C++最初的标准有何评价?

Lippman: 好的, 结果肯定会很有趣。我想说, C++标准的第一版, 是一次很糟糕的失败。他们把数学库放进去了, 接下去他们又后悔, 想要说服人们不要使用它们。异常处理也工作得不理想, constraint子句无法被真正地用在任何大型编程项目中, 用于确定实例化的Koenig查找规则无法正常工作。

换句话说, C++的第一版标准中有很大一部分无法正常工作。他们后来花费了巨大的努力, 才修复了大多数的问题。但我想强调一点, C++在当时还只是在一个很小的社区里流行, 其中有些人不惜代价而投身于此, 最终成为了真正的专家。

<http://www.csdn.net/article/2013-11-27/2817632-C++-Lippman>

下一篇 摘录的一些Bjarne Stroustrup关于C++的谈话内容

主题推荐      软件      语言      C++      《程序员》杂志      贝尔实验室

猜你在找

- |                                |                                   |
|--------------------------------|-----------------------------------|
| C++ Primer学习笔记24 C++中不可移植的语言特性 | 20110908 《C++ Primer》第一部分学习笔记汇总基本 |
| CC++程序员排序算法之标准C语言qsort函数简单用法   | C++语言发展历史 & 基础知识                  |
| Java程序员回头学C++ 之 C语言的标准输入输出     | 云时代C++语言应用开发的生存与发展                |
| Linux之父炮轰C++糟糕程序员的垃圾语言         | cc++语言总结C++ primer                |
| dos下使用的C++语言编程软件               | 索骥馆—编程语言之 《C++精髓软件工程方法》扫描         |

坚持1个月，听懂CNN

每天只要 5 分钟，坚持30天，效果真的不一样！

立即行动



仅限杭州

查看评论

暂无评论

发表评论

用户名:

zjufirefly

评论内容:

提交

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

aptech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap