O'REILLY®

HZ BOOKS
华章IT

Hack与HHVM权威指南

Hack and HHVM

Owen Yamauchi 著
苏南 译

机械工业出版社
China Machine Press

Hack与HHVM权威指南

Owen Yamauchi 著

苏南 译

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo O'REILLY®

O'Reilly Media, Inc.授权机械工业出版社出版

机械工业出版社

图书在版编目（CIP）数据

Hack 与 HHVM 权威指南 /（美）欧文·山内（Yamauchi，O.）著；苏南译 . —北京：机械工业出版社，2016.11

（O'Reilly 精品图书系列）

书名原文：Hack and HHVM: Programming Productivity Without Breaking Things

ISBN 978-7-111-55484-4

I. H… II. ①欧… ②苏… III. PHP 语言－程序设计－指南 IV. TP312-62

中国版本图书馆 CIP 数据核字（2016）第 289736 号

北京市版权局著作权合同登记

图字：01-2016-1903 号

书　　名 / Hack 与 HHVM 权威指南

书　　号 / ISBN 978-7-111-55484-4

责任编辑 / 缪杰

封面设计 / Ellie Volkhausen，张健

出版发行 / 机械工业出版社

地　　址 / 北京市西城区百万庄大街 22 号（邮政编码 100037）

印　　刷 / 北京市荣盛彩色印刷有限公司

开　　本 / 178 毫米 ×233 毫米　16 开本　15.5 印张

版　　次 / 2017 年 3 月第 1 版　2017 年 3 月第 1 次印刷

定　　价 / 69.00 元（册）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010)88379426；88361066

购书热线：(010)68326294；88379649；68995259

投稿热线：(010)88379604

读者信箱：hzit@hzbook.com

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

2014年3月21日，在PHP业界著名专家“鸟哥”（惠新宸）的新浪微博（@Laruence）下面，我第一次看到了Facebook公司的Hack语言及HHVM的相关消息。出于敏锐的职业嗅觉，我注册了相关的cn域名。在随后的周末，国内第一家Hack语言的中文站（*http://www.hacklang.org.cn*）和第一家HHVM中文站（*http://www.hhvm.com.cn*）正式出炉了。

一个月后在中关村的朔黄大厦，我很幸运地碰到了“鸟哥”，并和他畅谈了理想与人生。在鸟哥的启发下，我不久后就来到了新的工作岗位，开始了新的生活。环境改变了，生活和工作也踏上了新的征程。

2016年3月21日（很神奇，也是3月21日），机械工业出版社的缪杰编辑找到我，洽谈翻译本书的相关事宜。这正切合我内心深处由来已久的想法，即翻译一本专业的HHVM和Hack编程技术书籍，在中国范围内，传播最新的Hack语言编程技术知识。我非常强烈地感受到，必须抓住这个机会，完成这本书的翻译。

在2016年4月，我初步尝试翻译了前面的章节。5月到7月上旬，完成了本书大致的翻译。在工作之余以及周末的时间，我都在反复推敲英文原版书中的每个字句，经常熬夜到凌晨两点，这是一段难忘的经历。翻译同时也是不断学习提高的过程。在完成初稿后，我感觉又重新认识了Hack语言，也重新认识了PHP。

在2016年8月，刘诗灏编辑又给出了很多专业的修改意见。然后，整个译本又经历了几次脱胎换骨的变化，才最终呈现在广大读者的面前。

最后，我要感谢生活给我的磨砺。一份“不服输”的执念引领我在黑暗中不断探索前行。“不经历风雨，哪能见彩虹”，对此我有着更深刻的认识。我还要重点感谢缪杰和刘诗灏编辑。感谢他们在茫茫人海中发现我，让我能够有机会实现内心最深处的愿望：让Hack语言

编程技术在中国范围内得到更好的传播和发展。我还想感谢中国台湾作家张德芬，正是她的身心灵三部曲，伴我走过了那段最黑暗的时光，让我能够有机会学会接受、放下，学会通过“宇宙的力量”散发“心灵的喜悦”，最后达成“心想事成”。

Hack 语言和 PHP 的伴生与博弈不会停止，Hack 语言出身于 PHP，而又“高于”PHP 的特征，决定了这场争斗会旷日持久。在本书中，你可能会体会到 HHVM 及 Hack 语言的使用者对 PHP 的一些激烈言辞。同时，如果关注“鸟哥”的新浪微博，你也可以看到他对 Hack 语言和 HHVM 的一些反击之词。孰对孰错，见仁见智。

译者简介及联系方式

李苏南（笔名苏南），毕业于华北电力大学。浪迹京城十余载，地处中关村偏北。目前在国内某知名金融投资公司做构架相关的工作。如果您对我感兴趣，欢迎通过国内第一家 Hack 语言中文网站（*http://www.hacklang.org.cn*）找到我的联系方式。

由于经验不足等原因，本书的翻译过程不可避免地出现一些纰漏。如果您有疑问，欢迎访问 *http://www.hacklang.org.cn* 以及 *http://www.hhvm.com.cn*，参与我们的在线讨论和互动。同时，本书的相关源码及勘误表，也可以在上述网站上找到。

在本书出版时，实际的版本内容与书中相比，会不可避免地发生变化。我会在网站上及时地更新和调整。同时，本书中没有提及的“HHVM 安装过程”以及“支持 Hack 语言的编程工具”等内容，也会在网站中进行补充。

欢迎加入我们的讨论。与大家一起传播 Hack 语言，让更多的人体会到“更淋漓尽致的编程体验”。

苏南

2016 年 12 月

目录

序

2012 年，我和 Alok Menghrajan 开始了一项叫做“strict-mode”的新工作项目。简而言之，项目的目标就是在 HHVM 的基础之上构建一个静态类型版本的 PHP。

从此以后，我着迷于该项目所取得的成功，这个项目后来演变成大家所熟知的 Hack 语言。这个项目从最开始的类型检查器演变成为一门成熟的编程语言，并且有行业级别的工具做后盾支撑。

回望来路，当第一次向 HHVM 团队表达我们对项目的想法的时候，我有足够的理由相信他们认为我们疯了。但是最终，我们还是成功地说服了他们，让他们加入了我们的冒险之旅。

在 2012 年 6 月底的时候，Facebook 网站首次将 Hack 语言部署应用到产品中。就这样，没有任何管理上的认可，也没有任何形式的过程，Facebook 在生产中有了一门新的编程语言。

就在这个时候，我非常期待有人能够及时叫停我们，然而这样的情况却从未发生。

很多工程师跟随着我们的脚步。在我们知道之前，Facebook 内部大多数新的代码都使用 Hack 语言进行编写。于是我们决定把剩余的 PHP 代码库自动转为 Hack 语言的代码。我们从动态变量类型的 PHP 到静态变量类型的 Hack 语言，转化了巨大规模（上亿行）的代码。

这个过程是极具挑战性的，挑战主要来自于我们所采用的类型检查器的 C/S 技术框架，因为 PHP 开发人员已经习惯于一个快速编辑 / 刷新的周期，所以我们希望类型检查器有个快速的响应时间。这就是我们使用 Hack 语言类型检查服务器的原因：一个背后默默

支撑着类型信息的守护进程。当然，最为棘手的问题是如何保持与文件系统相一致的服务器状态。

为了使类型检查服务器更稳定地运行，我们度过了无数个不眠的夜晚。正是如此，也造就了 Hack 语言如此不平凡的今天。在面临海量代码更新的时候，类型检查器的响应时间几乎是瞬时的（快到可以自动完成）。

我非常高兴，现在有了这本 Hack 和 HHVM 方面的权威指导书。Owen 在此做了非常卓越的工作，甚至关于 Hack 语言的微妙之处，他都做了详细的解释。当然，还有其他一些需要你在 Hack/HHVM 的调试及生产过程中知晓的注意事项。

希望你能够享受使用 Hack 和 HHVM 的乐趣！

——Julien Verlaguet, Hack 语言之父

前言

在Facebook公司的大部分发展历程中，每隔几个月就会举办“黑客马拉松（hackathons）”活动，活动的目的在于鼓励工程师们碰撞出好的想法，而这些好的想法并不是和他们的日常工作相关的，他们自由组队，然后在一两天的时间内做出一些非常有意思的事情。

在2007年11月的一次“黑客马拉松”活动上，诞生了一个非常有意思的实验：一个工具能够将PHP程序转化为C++程序，然后还能够用C++编译器进行编译。想法是C++程序将会比PHP原生的程序运行起来快很多，因为它可以得益于多年以来对C++编译器的大量优化工作。

对于Facebook来说，这种可能性是非常有趣的，因为公司增加了大量新的用户，而支持更多新的用户需要耗费大量的CPU运算周期。所以当你耗尽所有可用的CPU运算周期后，除非你耗费大量财力购买更多的CPU，用来支持日益增多的用户所带来的CPU运算能力的需求，否则你必须寻找一个方法来降低每个用户的CPU消耗。由于Facebook整个网站的前端都是用PHP语言编写的，所以任何使PHP代码耗费更少CPU运算周期的新技术都受到欢迎。

在接下来的7年时间里，这个项目的发展远远超出了最开始在“黑客马拉松”中的起点。PHP到C++的转换器称为HPHPc， 在2009年的时候它成为支撑Fackbook网页业务唯一的服务器端引擎。在2010年年初，它以“HipHop for PHP”的名字开源了。然后从2010年起，一个全新的方法用来执行——即时编译为机器代码，并没有C++牵扯其中——脱胎换骨于HPHPc的代码库，并最终取代它。这个即时的编辑器称为“HipHop虚拟机”，简称为HHVM，并且在2013年的早期彻底取代了Facebook的网站服务器集群。早期的PHP到C++的转换器消失了，它没有在任何地方进行部署，同时它的代码都被删除了。

而 Hack 的起源是完全分开的，其根源在于试图在 PHP 中使用静态分析以自动探测潜在的安全漏洞的一个项目。很快，事实证明，PHP 的本质使得它在非常有用的静态分析方面很难有所进展。于是“严格模式（strictmode）”的想法就诞生了。对 PHP 进行修改，增加一些新的特性，比如引用、删除和添加一个补充的复杂类型系统。PHP 代码的作者可以自由选择是否使用严格模式，在保持完整的互操作性同时，获得更加强大的代码检查能力。

Hack 的方向从那时开始就作为基于 PHP 的类型系统掩盖了其本质。它在构建 Hack 编码的道路上获得了很多有重大影响的新特性，比如异步函数。它添加了很多包括集合在内的新特性，使得类型系统更加强大。本质上来说，它是一门和 PHP 不同的新语言，它已经在编程语言方面取得了自己的新位置。

以上就是 Hack 的发展历程，目前 Hack 是一门现代化的动态编程语言，它拥有鲁棒的静态类型检查能力，在 HHVM 上执行。HHVM 是一个和 PHP 无缝兼容且具有互操作性的实时编译运行时引擎。

什么是 Hack 和 HHVM

Hack 和 HHVM 是紧密联系在一起的，所以对于这些术语到底指代的是什么会有一些混乱。

Hack 是一门编程语言。它基于 PHP，继承了 PHP 中的很多语法，并且完全可以和 PHP 进行互操作。然而，很可能有人会认为 Hack 只是在 PHP 的基础上略加了装饰修改。Hack 最核心的特色是鲁棒的静态类型检查，这已经足够把 Hack 作为一门编程语言和 PHP 区分开了。对于现在已经从事已有 PHP 代码库开发方面工作的开发者来说，这是非常有益的。在这种情形下，将会给这些开发者很多的启迪，当然，对于新项目的底层开发也是一个非常不错的选择。

除了静态类型检查外，Hack 还拥有 PHP 没有的很多项新特性，本书将对这些新特性进行阐述：异步函数、XHP 等。出于解决一些粗糙边界问题的目的，Hack 也故意缺失了对一些 PHP 特性的支持。

HHVM 是一个执行引擎，它同时支持 PHP 和 Hack。它让两种语言可以互操作：PHP 书写的代码能够调用 Hack 代码，反之亦然。当执行 PHP 的时候，它的目标在于对 PHP.net 提供的 PHP 标准解释器进行替换。本书中有些章节的内容是关于 HHVM 的：如何配置并部署它，如何使用它调试和配置代码。

最后，我们要介绍的就是从 HHVM 中分离出来的 Hack 类型检查器：这是一个能够分析 Hack 代码（而不是 PHP 代码）然后报告类型错误的程序。在它能够接受的代码方面，

类型检查器目前要比 HHVM 严格一些。当然，在未来的发行版本中，HHVM 应该比类型检查器更加严格。目前，除了你在命令行里面启动它的命令“hh_client”外，类型检查器还没有个定型的名字，我更倾向于叫它“Hack 类型检查器”（Hack typechecker）或者就叫做“类型检查器”（typechecker）。

到目前为止，HHVM 是运行 Hack 的唯一执行引擎，这也是有时它们会混为一谈的原因。

本书读者对象

本书适合那些已经对编程有一定基础的读者。这里并没有花费时间解释很多编程语言里面常见的概念。例如控制流、数据类型、函数、面向对象编程等。

Hack 派生于 PHP，本书不会特别解释 PHP 中常见的语法，除非 Hack 里面相关的语法知识点与之不同。所以有 PHP 的知识基础将会非常有用。如果你从来没有使用过 PHP，但是有其他编程语言的相关经验，那么你仍然能够读懂本书里面的大部分代码。语法知识点都是非常易于理解的。

如果你拥有 PHP 相关经验，但是从未工作在一个复杂、高负载的 PHP 网站环境中，你也不必担心这里有什么你看不懂的。无论你的代码是简单独立运行的小脚本，还是数以百万行级别像 Facebook 一样的大型 Web 应用，Hack 对于任何规模的代码库都大有裨益。

这里有一些材料假设你对传统的 Web 应用已经熟悉，例如关系数据库查询、使用 memcached（见第 6 章）和生成 HTML（见第 7 章）。如果它们和你不相关，那么你可以跳过这些章节。但是事实上，对这些章节的理解并不需要什么特别的知识，哪怕是小的基础网站应用的开发经验。

我希望本书并不仅仅用来解释事物具体是什么的，而且希望介绍它是怎样的运作原理。程序语言设计是一个很困难的问题。它本质上是一门对上百个可能的方案统一进行权衡的艺术，同时它还受向后兼容的相关内容的制约，Hack 也不例外。所以如果你对“一门程序语言如何通过一系列不同寻常的限制来成就未来”这个案例学习感兴趣，那么本书将会提供你所需要的内容。

哲学理念

在 Hack 和 HHVM 设计的背后存在着一些理念。这些理念可以帮助你理解事物的运转方式。

程序类型

这里有个单独的观察程序用于指引 HHVM 优化和执行代码的方式，还指引 Hack 验证它的方式。它就是隐藏在大多数动态语言程序背后的一个静态类型的程序。

看一下如下代码，这段代码在 PHP 和 Hack 下都可以正常运行：

```
for ($i = 0; $i < 10; $i++) {
  echo $i + 100;
 }
```

虽然没有明确进行说明陈述，但是对于任何读者来说，很明显 `$i` 总是个整数。用计算机术语来说，`$i` 是单态（monomorphic）的：它只有一个类型。类型检查器将会利用这个属性来验证表达式“`$i + 100`”是否有意义。一个执行引擎也会利用这个属性把“`$i + 100`”编译为高效机器码，来做这个加法运算。

循环变量看起来似乎是个平常的例子，但事实证明，在现实世界的 PHP 代码库中，大多数的值都是单态的。这造成了一种直觉，对于一个值，如果不知道它的类型，你就不能做太多的事情。比如在它上面做算术运算，对它做索引，调用它上面的方法等。甚至在动态类型语言中，大多数的代码在使用它做任何事情之前，并不会检查它的类型，这就意味着这里有关于值类型的隐藏设定。如果运行的大多数代码不存在运行时类型错误，那么大多数的时候这些隐藏设定也必须成立。

HHVM 的方法是假设这个观点成立，并且相应地编译 PHP 和 Hack 为机器码。因为它运行的同时进行程序的编译，它知道将要编译的每条代码的逻辑流向。它将生成机器代码并假定这些类型：在前面所示的范例代码中，当编译表达式 `$i + 100` 时，HHVM 将看到 `$i` 是一个整数，然后使用单条加法硬件指令来做这项相加操作。

同时，Hack 的目的在于使隐藏的静态类型程序显示出来。它使一些类型进行显式的标通过标注使一些类型变成显式的，然后使用类型推理得到余下的类型信息。现在的理念就是，Hack 并不显式地限制已经存在的 PHP 程序，而是对 PHP 程序已经显露出来的行为进行鲁棒的静态分析。

这里有一点值得重复一下：Hack 的静态类型并不会带给你一个不同的编程风格。这门语言用来给你已经写好的程序一个更好的诠释方式。

逐步迁移

Hack 起源于一个数以百万行计的 PHP 代码库。不管这两种语言之间有多么相似，这里都没有什么办法能够一下子把如此大规模的代码库从一种语言迁移到另外一种语言，所以Hack采用了从PHP逐步迁移的路线。Hack 可以使用以PHP编写的函数和类，反之亦然。

对于 Hack 的每个特性，无论是否使用它的代码，这里都有无缝对接进行交互的方法。

另外，标准的 Hack/HHVM 发行版有自动从 PHP 到 Hack 迁移的工具，它还包含一个工具用于把 Hack 代码编译为 PHP 代码。这主要是为了方便一些类库的作者迁移到 Hack 上，同时为非HHVM用户保持一个使用其代码的通道。这些工具代码将在第 10 章中详细描述。

而 HHVM 致力于能够和标准的 PHP 解释器一样运行 PHP 代码。迁移 PHP 代码库到 Hack 的第一步就是要在 HHVM 上运行 PHP 代码。在这一步骤中，唯一重要的必要代码变更就是围绕扩展的，并不是所有的 PHP 和 Zend 扩展都兼容 HHVM。这里不应该由于核心语言的不同行为而导致扩展有所变化。

除去它的本源因素，对于开始一个新的项目，Hack 毫无疑问是个非常棒的选择。事实上，你会以这样的方式从 Hack 中获得最大的益处：当代码库 100% 是纯正的 Hack 代码时，这门语言会发挥它的极致。

本书是如何组织的

Hack 的核心功能是静态类型检查。它广泛涉及了 Hack 的所有其他特性，是 Hack 和 PHP 最显著的区别。本书第 1 章就详细地对本话题展开了阐述。本书所有其他章节的内容都依赖于对本章内容的理解，所以如果以前你没有接触过 Hack，我强烈建议你仔细阅读第 1 章的内容。第 2 章进行了补充，该章主要讨论了在 Hack 的类型体系中非常有趣的部分。

Hack 其余的特性彼此间几乎都是正交的。第 3 章主要解释了 Hack 的一些小特性。第 4 章展示了一些在 Hack 中并不存在的 PHP 特性以及这样的原因。第 5 章解释了如何及为什么使用 Hack 的集合类。第 6 章主要解释了 Hack 的多任务支持。第 7 章主要解释 Hack 中更加稳健、更加安全地创建 HTML 的语法和库。

第 8 章主要涉及设置、配置、部署和对 HHVM 的监控问题。第 9 章涉及 HHVM 的交互式调试器 hphpd。最后，第 10 章主要探索了一些编写 Hack 代码时所需要的工具，包括从 PHP 到 Hack 的迁移工具和一个交互式调试器。

版本

本书使用 Hack 3.6 和 HHVM 3.6，这个版本于 2015 年 3 月 11 日发布。(HHVM 和 Hack 的类型检查器存在于同一个代码库中，它们包含在同一个代码发行包内。) 当你读到这些内容的时候，应该会有更新的版本可用。然而，3.6 版本将会是个长期支持的版本。会在发布的 48 周后修复安全问题及 bug。

HHVM 3.6 实现了 PHP 5.6 的语义[注1]。它支持 PHP 5.6 中所有的新特性，包括常量 scalar 表达式、可变参数函数、取幂操作符等。这些新的特性在 Hack 3.6 上也存在。通常来说，当一个新版本的 PHP 发布时，HHVM 都会为 Hack 代码和 PHP 代码及时增加对新特性及语义的支持。

本书约定

本书中使用以下排版约定：

斜体

表明新术语、网址、电子邮件地址、文件名和文件扩展名。

等宽字体

用于代码清单以及在正文中引用程序元素，如变量或函数名、数据库、数据类型、环境变量、语句和关键字。

等宽粗体

显示用户应该输入的命令或其他文本。

这个元素表示提示或建议。

这个元素表示一般注解。

这个元素表示警告。

Safari® 图书在线

Safari Books Online 是一个随需而变的数字图书馆，在技术和商业方面以书和视频的形式，从世界领先的作者那里发布专业的内容。

专业技术人员、软件开发人员、网页设计人员、商业和创新型人才使用 Safari 图书在线，用于科研、解决问题、学习和资格培训。

注 1：相应的次要版本号仅仅是个巧合。一般，HHVM/Hack 和 PHP 版本号之间没有任何联系。