

一切可编译为 WebAssembly 的，
终将被编译为 WebAssembly。



WebAssembly 标准入门

—— 柴树杉 丁尔男 著 ——

张外借



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



WebAssembly 标准入门

—— 柴树彬 丁尔男 著 ——

人民邮电出版社
北京

图书在版编目 (CIP) 数据

WebAssembly标准入门 / 柴树杉, 丁尔男著. — 北京: 人民邮电出版社, 2019.1
ISBN 978-7-115-50059-5

I. ①W… II. ①柴… ②丁… III. ①编译软件 IV. ①TP314

中国版本图书馆CIP数据核字(2018)第251367号

内 容 提 要

WebAssembly 是一种新兴的网页虚拟机标准, 它的设计目标包括高可移植性、高安全性、高效率(包括载入效率和运行效率)、尽可能小的程序体积。本书详尽介绍了 WebAssembly 程序在 JavaScript 环境下的使用方法、WebAssembly 汇编语言和二进制格式, 给出了大量简单易懂的示例, 同时以 C/C++ 和 Go 语言开发环境为例, 介绍了如何使用其他高级语言开发 WebAssembly 模块。

本书适合从事高性能 Web 前端开发、跨语言 Web 应用开发的技术人员学习参考, 也可以作为 WebAssembly 标准参考手册随时查阅。

-
- ◆ 著 柴树杉 丁尔男
责任编辑 杨海玲
责任印制 焦志炜
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市君旺印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 13.25
字数: 246 千字
印数: 1—3 000 册
-

定价: 49.00 元

读者服务热线: (010) 81055410 印装质量热线: (010) 81055316

反盗版热线: (010) 81055315

广告经营许可证: 京东工商广登字 20170147 号

某一天，有朋友向我推荐了一项新技术——WebAssembly。我认为这是一项值得关注的技术。

说 WebAssembly 是一门编程语言，但它更像一个编译器。实际上它是一个虚拟机，包含了一门低级汇编语言和对应的虚拟机体系结构，而 WebAssembly 这个名字从字面理解就说明了一切——Web 的汇编语言。它的优点是文件小、加载快、执行效率非常高，可以实现更复杂的逻辑。

其实，我觉得出现这样的技术并不令人意外，而只是顺应了潮流，App 的封闭系统必然会被新一代 Web OS 取代。但现有的 Web 开发技术，如 JavaScript，前端执行效率和解决各种复杂问题的能力还不足，而 WebAssembly 的编译执行功能恰恰能弥补这些不足。WebAssembly 标准是在谋智（Mozilla）、谷歌（Google）、微软（Microsoft）、苹果（Apple）等各大厂商的大力推进下诞生的，目前包括 Chrome、Firefox、Safari、Opera、Edge 在内的大部分主流浏览器均已支持 WebAssembly。这使得 WebAssembly 前景非常好。

WebAssembly 是 Web 前端技术，具有很强的可移植性，技术的潜在受益者不局限于传统的前端开发人员，随着技术的推进，越来越多的其他语言的开发者也将从中受益。如果开发者愿意，他们可以使用 C/C++、Go、Rust、Kotlin、C# 等开发语言来写代码，然后编译为 WebAssembly，并在 Web 上执行，这是不是很酷？它能让我们很容易将用其他编程语言编写的程序移植到 Web 上，对于企业级应用和工业级应用都是巨大利好。

WebAssembly 的应用场景也相当丰富，如 Google Earth，2017 年 10 月 Google Earth 开始在 Firefox 上运行，其中的关键就是使用了 WebAssembly；再如网页游戏，WebAssembly 能让 HTML5 游戏引擎速度大幅提高，国内一家公司使用 WebAssembly 后引擎效率提高了 300%。

WebAssembly 作为一种新兴的技术，为开发者提供了一种崭新的思路和工作方式，未来是很有可能大放光彩的，不过目前其相关的资料和社区还不够丰富，尽管已经有一些社区开始出现了相关技术文章，CSDN 上也有较多的文章，但像本书这样全面系统地介绍 WebAssembly 技术的还不多，甚至没有。本书的两位作者都是有 10 多年经验的一线开发者，他们从 WebAssembly 概念诞生之初就开始密切关注该技术的发展，其中柴树

杉是 Emscripten (WebAssembly 的技术前身之一) 的首批实践者, 丁尔男是国内首批工程化使用 WebAssembly 的开发者。

2018 年 7 月, WebAssembly 社区工作组发布了 WebAssembly 1.0 标准。现在, 我在第一时间就向国内开发者介绍和推荐本书, 是希望开发者能迅速地了解和 Learning 新技术, 探索新技术的价值。

蒋涛

CSDN 创始人、总裁, 极客帮创始合伙人

本书结构

第 0 章回顾了 WebAssembly 的诞生背景。

第 1 章针对不熟悉 JavaScript 的读者，介绍本书将使用到的部分 JavaScript 基础知识，包括 console 对象、箭头函数、Promise、ArrayBuffer 等。对 JavaScript 很熟悉的读者可以跳过本章。

第 2 章通过两个简单的小例子展示 WebAssembly 的基本用法。

第 3 章和第 4 章分别从外部和内部两个角度详细介绍 WebAssembly，前者着重于相关的 JavaScript 对象，后者着重于 WebAssembly 虚拟机的内部运行机制。因为 WebAssembly 跨越了两种语言、两套运行时结构，所以读者阅读第 3 章时可能会感到不明就里——为什么多数指令中没有操作数？所谓的“栈式虚拟机”到底是什么？类似的疑问都将在第 4 章中得到解答。在写作本书时，我们期望读者读完第 4 章后复读第 3 章时，能有豁然开朗的感觉。

第 5 章介绍 WebAssembly 汇编的二进制格式。若想尝试自己实现 WebAssembly 虚拟机或者其他语言到 WebAssembly 的编译器，掌握二进制汇编格式是必需的。即使不开展类似的项目，通过阅读本章也可以加深对 WebAssembly 虚拟机架构的整体认识，厘清各种全局索引的相互关系。

第 6 章和第 7 章分别以 C/C++ 和 Go 语言为例，介绍如何使用高级语言来开发 WebAssembly 应用。WebAssembly 全手工编写 .wat 文件实现大型模块的机会并不会很多。在实际工程中，WebAssembly 作为一门类汇编语言，更多的是作为其他语言的编译目标而存在。目前 C/C++、Rust、Go、Lua、Kotlin、C# 均已支持 WebAssembly，可以预见这一支持列表将越来越长。

附录列出了现有的 200 多条 WebAssembly 指令及其作用。

致谢

感谢蒋涛先生为本书作序，感谢所有为 WebAssembly 标准诞生作出努力的朋友。其中特别感谢 Emscripten 和 asm.js 的作者，没有他们的灵感，WebAssembly 标准就不可能诞生。感谢 WebAssembly 工作组的专家，是他们的工作让我们看到了草案 1.0 的成果。感谢各大浏览器厂商为 WebAssembly 提供的支持。最后，感谢人民邮电出版社的杨海玲编辑，没有她，本书就不可能出版。谢谢大家！

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书提供书中的源代码，要获得以上配套资源，请在异步社区本书页面中点击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意：为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，点击“提交勘误”，输入勘误信息，点击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

详细信息

写书评

提交勘误

页码: 页内位置 (行数): 勘误次数:

B I U

字数统计

提交

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异

步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术。



异步社区



微信服务号

目 录

第0章 WebAssembly 诞生背景	1
0.1 JavaScript 简史	1
0.2 asm.js 的尝试	2
0.3 WebAssembly 的救赎	5
第1章 JavaScript 语言基础	7
1.1 console 对象	7
1.2 函数和闭包	9
1.3 Promise 对象	12
1.4 二进制数组	13
第2章 WebAssembly 快速入门	17
2.1 准备工作	17
2.1.1 WebAssembly 兼容性	17
2.1.2 WebAssembly 文本格式与 wabt 工具集	19
2.2 首个例程	21
2.3 WebAssembly 概览	22
2.3.1 WebAssembly 中的关键概念	23
2.3.2 WebAssembly 程序生命周期	24
2.3.3 WebAssembly 虚拟机体系结构	25
2.4 你好, WebAssembly	25
2.4.1 WebAssembly 部分	26
2.4.2 JavaScript 部分	27
2.5 WebAssembly 调试及代码编辑环境	28

第3章 JavaScript 中的 WebAssembly 对象	31
3.1 WebAssembly 对象简介	31
3.2 全局方法	32
3.2.1 WebAssembly.compile()	32
3.2.2 WebAssembly.instantiate()	33
3.2.3 WebAssembly.validate()	34
3.2.4 WebAssembly.compileStreaming()	35
3.2.5 WebAssembly.instantiateStreaming()	35
3.3 WebAssembly.Module 对象	36
3.3.1 WebAssembly.Module()	36
3.3.2 WebAssembly.Module.exports()	37
3.3.3 WebAssembly.Module.imports()	38
3.3.4 WebAssembly.Module.customSections()	39
3.3.5 缓存 Module	40
3.4 WebAssembly.Instance 对象	41
3.4.1 WebAssembly.Instance()	41
3.4.2 WebAssembly.Instance.prototype.exports	42
3.4.3 创建 WebAssembly.Instance 的简洁方法	43
3.5 WebAssembly.Memory 对象	44
3.5.1 WebAssembly.Memory()	44
3.5.2 WebAssembly.Memory.prototype.buffer	44
3.5.3 WebAssembly.Memory.prototype.grow()	47
3.6 WebAssembly.Table 对象	50
3.6.1 WebAssembly.Table()	51
3.6.2 WebAssembly.Table.prototype.get()	52
3.6.3 WebAssembly.Table.prototype.length	54
3.6.4 在 WebAssembly 内部使用表格	55
3.6.5 多个实例通过共享表格及内存协同工作	57
3.6.6 WebAssembly.Table.prototype.set()	60
3.6.7 WebAssembly.Table.prototype.grow()	61
3.7 小结及错误类型	61

第4章 WebAssembly 汇编语言	65
4.1 S-表达式	65
4.2 数据类型	66
4.3 函数定义	67
4.3.1 函数签名	67
4.3.2 局部变量表	68
4.3.3 函数体	68
4.3.4 函数别名	68
4.4 变量	69
4.4.1 参数与局部变量	69
4.4.2 变量别名	70
4.4.3 全局变量	70
4.5 栈式虚拟机	72
4.5.1 栈	72
4.5.2 WebAssembly 栈式虚拟机	72
4.5.3 栈式调用	73
4.6 函数调用	75
4.6.1 直接调用	75
4.6.2 间接调用	76
4.6.3 递归	78
4.7 内存读写	79
4.7.1 内存初始化	79
4.7.2 读取内存	80
4.7.3 写入内存	81
4.7.4 获取内存容量及内存扩容	82
4.8 控制流	83
4.8.1 nop 和 unreachable	83
4.8.2 block 指令块	83
4.8.3 if 指令块	85
4.8.4 loop 指令块	86
4.8.5 指令块的 label 索引及嵌套	86
4.8.6 br	87

4.8.7	br_if	89
4.8.8	return	90
4.8.9	br_table	90
4.9	导入和导出	91
4.9.1	导出对象	91
4.9.2	导入对象	93
4.10	start() 函数及指令折叠	96
4.10.1	start() 函数	96
4.10.2	指令折叠	97
第 5 章	WebAssembly 二进制格式	99
5.1	LEB128 编码	99
5.1.1	LEB128 编码原理	99
5.1.2	无符号数的 LEB128 编码	100
5.1.3	有符号数的 LEB128 编码	101
5.2	头部和段数据	101
5.2.1	头部	101
5.2.2	段类型列表	101
5.2.3	段数据结构	102
5.3	内存段和数据段	104
5.3.1	内存段	104
5.3.2	数据段	105
5.4	表格段和元素段	106
5.4.1	表格段	107
5.4.2	元素段	108
5.5	开始段和函数索引	108
5.5.1	开始段	109
5.5.2	函数索引	110
5.6	全局段	111
5.6.1	全局变量索引	111
5.6.2	全局段编码方式	112
5.7	函数段、代码段和类型段	113
5.7.1	函数段、代码段和类型段之间的关系	113

5.7.2 更简单的例子	114
5.7.3 函数段	115
5.7.4 类型段	116
5.7.5 代码段	117
5.8 导入段和导出段	118
5.8.1 例子	118
5.8.2 类型段	119
5.8.3 导入段	119
5.8.4 导出段	120
5.9 自定义段	121
5.9.1 自定义段结构	122
5.9.2 模块的名字	122
5.9.3 全局变量的名字	123
5.9.4 函数的名字	124
5.9.5 局部变量的名字	124
5.10 补充说明	126
第 6 章 Emscripten 和 WebAssembly	127
6.1 安装环境	127
6.1.1 emsdk 命令安装	127
6.1.2 Docker 环境安装	128
6.1.3 验证 emcc 命令	129
6.2 你好, Emscripten!	129
6.2.1 生成 wasm 文件	130
6.2.2 浏览器环境	130
6.2.3 自动生成 HTML 测试文件	132
6.3 C/C++内联 JavaScript 代码	133
6.3.1 EM_ASM 宏	133
6.3.2 EM_ASM_宏	134
6.3.3 EM_ASM_*宏	135
6.3.4 函数参数	137
6.3.5 注意问题	138
6.4 C/C++调用 JavaScript 函数	139

6.4.1 C 语言版本的 eval() 函数	139
6.4.2 打造带参数的 eval() 函数	141
6.4.3 打造可变参数的 eval() 函数	143
6.4.4 eval() 函数返回字符串	146
6.5 JavaScript 调用 C 导出函数	149
6.5.1 调用导出函数	149
6.5.2 辅助函数 ccall() 和 cwrap()	152
6.6 运行时和消息循环	155
6.6.1 Emscripten 运行时	155
6.6.2 消息循环	158
6.7 补充说明	162
第 7 章 Go 语言和 WebAssembly	163
7.1 你好, Go 语言	163
7.2 浏览器中的 Go 语言	166
7.3 使用 JavaScript 函数	168
7.4 回调 Go 函数	170
7.5 syscall/js 包	172
7.6 WebAssembly 模块的导入函数	175
7.7 WebAssembly 虚拟机	178
7.8 补充说明	180
附录 指令参考	181

第 0 章

WebAssembly 诞生背景

一切可编译为 WebAssembly 的，终将被编译为 WebAssembly。

——Ending

WebAssembly 是一种新兴的网页虚拟机标准，它的设计目标包括：高可移植性、高安全性、高效率（包括载入效率和运行效率）、尽可能小的程序体积。

0.1 JavaScript 简史

只有了解了过去才能理解现在，只有理解了现在才可能掌握未来的发展趋势。JavaScript 语言因为互联网而生，紧随着浏览器的出现而问世。JavaScript 语言是 Brendan Eich 为网景（Netscape）公司的浏览器设计的脚本语言，据说前后只花了 10 天的时间就设计成型。为了借当时的明星语言 Java 的东风，这门新语言被命名为 JavaScript。其实 Java 语言和 JavaScript 语言就像是雷锋和雷锋塔一样没有什么关系。

JavaScript 语言从诞生开始就是严肃程序员鄙视的对象：语言设计垃圾、运行比蜗牛还慢、它只是给不懂编程的人用的玩具等。当然出现这些观点也有一定的客观因素：JavaScript 运行确实够慢，语言也没有经过严谨的设计，甚至没有很多高级语言标配的块作用域特性。

但是到了 2005 年，Ajax（Asynchronous JavaScript and XML）方法横空出世，JavaScript 终于开始火爆。据说是 Jesse James Garrett 发明了这个词汇。谷歌当时发布的 Google Maps