

# 深入浅出WebAssembly



[深入浅出WebAssembly\\_下载链接1](#)

著者:于航

出版者:电子工业出版社

出版时间:2018-11

装帧:平装

isbn:9787121352171

WebAssembly是一种新的二进制格式，它可以方便地将C/C++等静态语言的代码快速地“运行”在浏览器中，这一特性为前端密集计算场景提供了无限可能。不仅如此，通过

WebAssembly技术，我们还可以将基于Unity等游戏引擎开发的大型游戏快速地移植到Web端。WebAssembly技术现在已经被计划设计成W3C的标准，众多浏览器厂商已经提供了对其MVP版本标准的支持。在Google I/O 2017大会上，Google首次针对WebAssembly技术进行了公开演讲和推广，其Post-MVP版本标准更是对诸如DOM操作、多线程和GC等特性提供了支持。WebAssembly所带来的Web技术变革势不可挡。

《深入浅出WebAssembly》力求从一些简单的实践入手，深入理论，到复杂的具有实际业务价值的综合实践，深入浅出地介绍Wasm技术发展至今，其背后所涉及的各种底层设计原理与实现、相关工具链以及未来发展方向等多方面内容。本书内容包括：WebAssembly技术的发展历程，从PNaCl到ASM.js再到WebAssembly，以及这些技术的基本应用方法与性能对比；WebAssembly的标准上层API、底层堆栈机的设计原理，以及对MVP标准理论的深入解读；与WebAssembly标准相关的进阶内容，如单指令多数据流（SIMD）、动态链接（DL）等；LLVM工具链与WAT可读文本格式的相关内容；基于Emscripten工具链开发WebAssembly应用的基本流程，以及工具链的一些基本常用功能和特性；基于Emscripten工具链实现C/C++语言动态关系绑定技术；Emscripten工具链所提供的一些如WebGL支持、虚拟文件系统、应用优化以及HTML5事件系统等高级应用特性；构建一个具有实际业务价值的WebAssembly应用，现阶段Wasm生态的发展情况，以及在Post-MVP标准中制订的一些WebAssembly未来发展规划。

《深入浅出WebAssembly》的目标读者为Web前端开发人员、C/C++开发人员和WebAssembly技术感兴趣的人员。

作者介绍：

于航，现就职于阿里巴巴 / 饿了么事业部（BU），资深前端工程师，FreeCodeCamp(FCC) China 上海社区负责人，QCon(2017)上海前端专场讲师。2016年开始研究WebAssembly技术，2017年加入WebAssembly中国社区，同年加入官方WCG（W3C Community Group），定期参与WCG组织的各种线上视频研讨会议，在跟进WebAssembly新发展的同时，也为WebAssembly的标准化提出自己的建议和意见。生活中喜欢弹钢琴、演讲与分享。主要技术研究方向为Web前端与基础设施架构、WebAssembly、LLVM以及编译器等相关领域。

目录: 第1章 漫谈WebAssembly发展史	1
1.1 JavaScript的发展和弊端	1
1.1.1 快速发展与基准测试	1
1.1.2 Web新时代与不断挑战	8
1.1.3 无法跨越的“阻碍”	11
1.1.4 Chrome V8引擎链路	17
1.2 曾经尝试——ASM.js与PNaCl	28
1.2.1 失落的ASM.js	28
1.2.2 古老的NaCl与PNaCl	42
1.3 新的可能——WebAssembly	57
1.3.1 改变与颠覆	57
1.3.2 一路向前，WCG与WWG	85
第2章 WebAssembly核心原理（基于MVP标准）	90
2.1 应用与标准Web接口	90
2.1.1 编译与初始化	90
2.1.2 验证模块	106
2.1.3 遇到错误	106

- 2.1.4 内存分配 108
- 2.1.5 表 112
- 2.2 深入设计模型——堆栈机 118
  - 2.2.1 堆栈式虚拟机 119
  - 2.2.2 逆波兰表达式 125
  - 2.2.3 Shunting-yard算法 126
  - 2.2.4 标签与跳转 130
  - 2.2.5 条件语句 135
  - 2.2.6 子程序调用 137
  - 2.2.7 变量 138
  - 2.2.8 栈帧 139
  - 2.2.9 堆 140
- 2.3 类型检查 141
  - 2.3.1 数据指令类型 142
  - 2.3.2 基本流程控制 144
  - 2.3.3 基于表达式的控制流 149
  - 2.3.4 类型堆栈的一致性 151
  - 2.3.5 不可达代码 155
- 2.4 二进制编码 156
  - 2.4.1 字节序——大端模式与小端模式 157
  - 2.4.2 基于LEB-128的整数编码 161
  - 2.4.3 基于IEEE-754—2008的浮点数编码 164
  - 2.4.4 基于UTF-8的字符串编码 167
  - 2.4.5 模块数据类型 168
  - 2.4.6 虚拟指令与编码 169
  - 2.4.7 类型构造符 174
- 2.5 模块 175
  - 2.5.1 段 175
  - 2.5.2 索引空间 185
  - 2.5.3 二进制原型结构 186
- 2.6 内存结构 196
  - 2.6.1 操作运算符 197
  - 2.6.2 寻址 197
  - 2.6.3 对齐 198
  - 2.6.4 溢出与调整 202
- 第3章 动态链接与SIMD（基于MVP标准） 204
  - 3.1 动态链接（Dynamic Linking） 204
    - 3.1.1 ELF 206
    - 3.1.2 重定向（Relocation） 212
    - 3.1.3 GOT（Global Offset Table, 全局偏移表） 225
    - 3.1.4 PLT（Procedure Lookup Table, 过程查询表） 229
    - 3.1.5 基于表的Wasm模块动态链接 233
  - 3.2 单指令多数数据流（SIMD） 236
    - 3.2.1 SIMD应用 238
    - 3.2.2 并行与并发 243
    - 3.2.3 费林分类法 244
    - 3.2.4 SIMD.js & TC39 246
    - 3.2.5 WebAssembly上的SIMD扩展 248
- 第4章 深入LLVM与WAT 250
  - 4.1 LLVM——底层虚拟机 250
    - 4.1.1 传统的编译器架构 251
    - 4.1.2 LLVM中间表示层 252
    - 4.1.3 基于LLVM的编译器架构 254
    - 4.1.4 LLVM优化策略 256

- 4.1.5 LLVM命令行工具 261
- 4.1.6 WebAssembly与LLVM 267
- 4.2 基于LLVM开发编程语言 272
  - 4.2.1 图灵完备与DSL 276
  - 4.2.2 简易词法分析器 280
  - 4.2.3 RDP与OPP算法 287
  - 4.2.4 AST 295
  - 4.2.5 简易语法分析器 296
  - 4.2.6 生成LLVM-IR代码 303
  - 4.2.7 链接优化器 307
  - 4.2.8 编译到目标代码 308
  - 4.2.9 整合I/O交互层 312
- 4.3 WAT 315
  - 4.3.1 S-表达式 317
  - 4.3.2 WAT/Wasm与Binary-AST 319
  - 4.3.3 其他与设计原则 320
- 第5章 Emscripten基础应用 322
  - 5.1 利器——Emscripten工具链 322
    - 5.1.1 Emscripten发展历史 322
    - 5.1.2 Emscripten组成结构 324
    - 5.1.3 Emscripten下载、安装与配置 326
    - 5.1.4 运行测试套件 330
    - 5.1.5 编译到ASM.js 331
  - 5.2 连接C/C++与WebAssembly 333
    - 5.2.1 构建类型 333
    - 5.2.2 Emscripten运行时环境 342
    - 5.2.3 在JavaScript代码中调用C/C++函数 351
    - 5.2.4 在C/C++代码中调用JavaScript函数 362
- 第6章 基于Emscripten的语言关系绑定 382
  - 6.1 基于Embind实现关系绑定 384
    - 6.1.1 简单类 389
    - 6.1.2 数组与对象类型 391
    - 6.1.3 高级类元素 393
    - 6.1.4 重载函数 407
    - 6.1.5 枚举类型 408
    - 6.1.6 基本类型 409
    - 6.1.7 容器类型 411
    - 6.1.8 转译JavaScript代码 413
    - 6.1.9 内存视图 416
  - 6.2 基于WebIDL实现关系绑定 417
    - 6.2.1 指针、引用和值类型 420
    - 6.2.2 类成员变量 422
    - 6.2.3 常量“const”关键字 423
    - 6.2.4 命名空间 424
    - 6.2.5 运算符重载 425
    - 6.2.6 枚举类型 426
    - 6.2.7 接口类 429
    - 6.2.8 原始指针、空指针与void指针 430
    - 6.2.9 默认类型转换 433
- 第7章 探索Emscripten高级特性 436
  - 7.1 加入优化流程 436
    - 7.1.1 使用编译器代码优化策略 441
    - 7.1.2 使用GCC压缩代码 443
    - 7.1.3 使用IndexedDB缓存模块对象 445

- 7.1.4 其他优化参数 452
- 7.2 使用标准库与文件系统 453
  - 7.2.1 使用基于musl和libc++的标准库 454
  - 7.2.2 虚拟文件系统结构 457
  - 7.2.3 打包初始化文件 458
  - 7.2.4 基本文件系统操作 460
  - 7.2.5 懒加载 469
  - 7.2.6 Fetch API 472
- 7.3 处理浏览器事件 477
  - 7.3.1 事件注册函数 479
  - 7.3.2 事件回调函数 479
  - 7.3.3 通用类型与返回值类型 481
  - 7.3.4 常用事件 482
- 7.4 基于EGL、OpenGL、SDL和OpenAL的多媒体处理 485
  - 7.4.1 使用EGL与OpenGL处理图形 486
  - 7.4.2 使用SDL处理图形 492
  - 7.4.3 使用OpenAL处理音频 495
- 7.5 调试WebAssembly应用 497
  - 7.5.1 编译器的调试信息 498
  - 7.5.2 使用调试模式 500
  - 7.5.3 手动跟踪 501
  - 7.5.4 其他常用编译器调试选项 503
- 第8章 WebAssembly综合实践、发展与未来 504
  - 8.1 DIP综合实践应用 504
    - 8.1.1 应用描述 504
    - 8.1.2 滤镜与卷积 505
    - 8.1.3 基本组件类型与架构 509
    - 8.1.4 编写基本页面骨架 (HTML与CSS) 510
    - 8.1.5 编写核心卷积函数 (C++) 511
    - 8.1.6 编写主渲染循环与“胶水”代码 (JavaScript) 513
    - 8.1.7 使用Emscripten编译并运行应用 518
    - 8.1.8 性能对比 519
  - 8.2 WebAssembly常用工具集 520
    - 8.2.1 Cheerp 520
    - 8.2.2 Webpack 4 522
    - 8.2.3 Go和Rust的WebAssembly实践 525
    - 8.2.4 Binaryen 527
    - 8.2.5 WasmFiddle 528
    - 8.2.6 Wabt 529
    - 8.2.7 AssemblyScript 529
  - 8.3 WebAssembly未来草案 529
    - 8.3.1 GC (垃圾回收) 530
    - 8.3.2 Multi-Thread (多线程) 与原子操作 530
    - 8.3.3 异常处理 530
    - 8.3.4 多返回值扩展 530
    - 8.3.5 ES模块 530
    - 8.3.6 尾递归 531
    - 8.3.7 BigInts的双向支持 531
    - 8.3.8 自定义注释语法 531
    - • • • • [\(收起\)](#)

[深入浅出WebAssembly\\_下载链接1](#)

## 标签

WebAssembly

前端开发

前端

C++

计算机

计算机科学

编程

前端技术

## 评论

这是今年看过的最尴尬的一本书，这本书从指令分类的概念讲到elf的基础，C++的基础，编译器的基础，解释器的基础，LLVM的基础。。。基本上就是给大一新生在写教材。WebAssembly标准只有150页，不知道作者在写些什么东西，浪费时间。

-----  
又遇到一个秀知识面的作者，这种书你都说不上是好是坏，毕竟能够写那么多字也是不容易

-----  
太泛，着重于wasm本身的讨论太少。

-----  
作者很有诚意

-----  
WebAssembly整套编译流程打通。作者大概担心读者基础不足，补充了大量的基础知识，讲的挺好的书。当然如果已经有相应领域知识的话可能会略显啰嗦了

-----  
前面写的跟wa一点关系没有，到wa的部分零零散散，对入门者极不友好

-----  
适合进阶

-----  
读了一小部分，作者从浏览器底层和asm.js娓娓道来，还没进入主题就科普了许多相关干货，不过读了两章就啃不太动了，缺的知识有点多

-----  
[深入浅出WebAssembly\\_下载链接1](#)

## 书评

@c623fc836f3be 1. 这位朋友说整本书讲了很多基础，各种基础。作为国内第一本介绍Wasm技术的书籍，我不能上来就讲VM的原理。堆栈机模型基础是理解Wasm字节码的基础，所以是必不可少的部分；从基础讲起也是为了降低读者阅读此书的难度。 2. 我并没有写C++基础，如果你认...

-----  
[深入浅出WebAssembly\\_下载链接1](#)