

WebKit 技术内幕

朱永盛 著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

WebKit 技术内幕

朱永盛 著



电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书从炙手可热的 HTML5 的基础知识入手,重点阐述目前应用最广的渲染引擎项目——WebKit。不仅着眼于系统描述 WebKit 内部渲染 HTML 网页的原理,并基于 Chromium 的实现,阐明渲染引擎如何高效地利用硬件和最新技术,而且试图通过对原理的剖析,向读者传授实现高性能 Web 前端开发所需的宝贵经验。

全书首先从总体上描述 WebKit 架构和组成,而后涵盖 Web 前端和所有与之相关的重要技术,包括网络、资源加载、HTML 和 CSS 解析、渲染树、布局、硬件加速、JavaScript 引擎、多媒体、移动支持、插件机制、安全机制、调试和最新的 Web 平台等。对于每一项技术,在介绍基本含义之上,详细分析 WebKit 内部的工作原理,进而从实践角度道出由此带来的 Web 前端开发启示。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

WebKit 技术内幕 / 朱永盛著. —北京: 电子工业出版社, 2014.6
ISBN 978-7-121-22964-0

I. ①W... II. ①朱... III. ①网页制作工具—程序设计 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2014)第 075051 号

策划编辑: 张春雨

责任编辑: 牛 勇

印 刷: 北京中新伟业印刷有限公司

装 订: 北京中新伟业印刷有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16

印张: 28.5 字数: 501.6 千字

印 次: 2014 年 6 月第 1 次印刷

印 数: 3000 册 定价: 79.00 元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前言

随着 HTML5 的快速发展和网络时代的到来，Web 的接入口——浏览器越来越重要，而作为浏览器的内核——渲染引擎也变成了热门话题。自笔者接触 HTML5 技术和浏览器以来，深深地被这一包含众多非凡技术的新领域所吸引，并由此产生了很多疑问，为此，我开始了漫长的学习和研究 WebKit（及 Blink）渲染引擎和 Chromium 浏览器的征程。虽然 WebKit 项目本身非常复杂，但是其简单的代码结构、清晰的逻辑给我留下了深刻的印象，因为在这些复杂技术的背后，竟然也可以使用良好的设计去解决技术的复杂性。而基于 WebKit 的 Chromium 项目更是将众多大胆的新技术引入到了浏览器领域，让人耳目一新。

WebKit 是一个非常成功的项目，它不仅仅是个渲染引擎，而且成功地推动了网络的发展。基于 WebKit 渲染引擎的浏览器项目 Chromium，更是成为率先支持 HTML5 功能和创新新功能的标杆。要完整理解一个 Web 渲染引擎和浏览器并不容易，因为它们的确包含了众多复杂的功能。据笔者的统计，WebKit 项目和 Chromium 项目（不包括该项目依赖的众多第三方项目）的代码量都在 500 万行以上，而这些代码很多并没有完善的文档，所以理解这些技术背后的工作原理还是非常困难的。

随着学习的深入，笔者发现目前对于整个渲染引擎的分析和文档化还处于一个缺失的状态。同时，因为渲染引擎和浏览器包含了太多的技术，让人有点应接不暇的感觉。虽然 WebKit 项目代码结构简单，但是由于文档的缺失，爱好者对于每一项

新技术，也经常有不知从何下手的感觉。为此，笔者结合自身的理解，通过这本书系统性地分析这一领域的众多技术，希望能帮助读者快速度过迷茫的时期。

本书的读者

本书主要是为 Web 爱好者准备的一本书，主要针对 Web 前端开发者、浏览器开发者、Web 平台开发者和一切对 HTML5 技术、WebKit 渲染引擎和 Chromium 浏览器的工作原理感兴趣的读者。对于 Web 前端开发者而言，笔者一直认为，如果使用 HTML5 技术来编写网页或者 Web 应用，了解其背后的工作原理是写出高效代码的有效捷径。就像开发者想编写高效 C++ 代码，需要理解 C++ 编译器背后的原理一样，因为只有这样，开发者才能够编写出高性能的代码。对于浏览器开发者来说，本书着重介绍现在非常热门的 WebKit（及 Blink）渲染引擎和非常先进的 Chromium 浏览器，通过解释其内部的工作机制和原理，让开发者可以很快理解这一切的前因后果。对于其他的广大爱好者来说，HTML5 技术才刚刚开始，未来的发展还将继续，了解这一技术有助于扩展视野，而且理解浏览器对各种技术的应用和设计，对于大家理解很多其他领域的技术也有很强的启发作用。

因为本书的介绍主要是基于对 WebKit 和 Chromium 内部原理解释来进行，而这些项目也都是基于 C/C++ 代码来编写，所以读者最好对该语言有一些了解。不过，如果不了解它也没有太大的关系，只要对面向对象编程的思想有所了解，阅读本书也没有太大的障碍。同时，本书不是一本介绍编写 HTML/JavaScript 代码的书，所以，不会对 HTML 的编程做过多详细的解释，而是以一种简单的方式描述一些基础性常识。

本书的组织

本书基本的写作方式是力求在介绍 HTML5 技术的基础上，通过对 W3C 组织制

定的规范的解释，进一步解读 WebKit 渲染引擎和 Chromium 浏览器是如何设计出高效的架构来支持这些 HTML5 技术规范的，其中着重剖析内部的框架和工作原理。在很多情况下，笔者也试图通过一些开发和工作实践来帮助理解这些框架和实现背后的机制和原理。

如果想了解整个渲染引擎的原理，光靠渲染引擎本身不足以说明所有机制，所以本书自始至终都是结合 WebKit 项目和基于 WebKit 的 Chromium 浏览器项目来描述其工作原理的，因为 WebKit 项目本身不是一个浏览器，而 Chromium 浏览器的设计和架构可以帮助读者完整理解网页的渲染过程和现代 HTML5 新技术是如何获得支持的，这一过程的确非常精彩。

为了理解 HTML5 新技术和浏览器的工作原理，本书着重带来以下方面的详细分析，包括 HTML5 技术分析、渲染引擎和浏览器介绍、WebKit 渲染引擎框架、Chromium 框架和进程架构、网页和网页结构、渲染过程、网络栈、HTML 语言、DOM、CSS 样式、布局计算、渲染基础、高级硬件加速机制、JavaScript 引擎、插件和扩展、多媒体、移动领域、安全机制、调试机制、发展趋势和 Web 平台等众多热门技术和前沿性话题。笔者希望将 HTML5 中绝大多数的重要技术都展现出来，让读者可以对这个领域的众多技术有个总体把握并对主要技术的前因后果有较为深入的理解。

本书引用的参考资料都是笔者多年来研究的对象，对于笔者理解 HTML5 技术、前端开发技术、渲染引擎和浏览器技术起了非常重要的作用，一些论题可能在本书中介绍得不够完善，读者可以参考这些资料，做进一步的学习和研究。

本书是一个讲解内部原理的书，涉及众多的技术，特别是深入技术内部工作机制的地方，由于这些内容非常复杂，而且是根据笔者个人的理解加以分析，所以很多时候可能存在理解上的偏差或者错误。如果有什么不妥之处，还望广大读者谅解并给予指导，或者将意见发送到我的邮箱：yongsheng@chromium.org。

致谢

感谢电子工业出版社的张春雨、王新宇、尚冰雪等编辑，自始至终给予笔者的强有力的帮助和支持。特别感谢我在英特尔亚太研发有限公司的同事，包括但不限于闵洪波、王兴楠、余枝强、刘守群、朱俊敏、王视臻、胡宁馨、高纯、尹立、顾扬、冯海涛、霍海涛等，他们同我一起探讨了很多关于 HTML5、WebKit 和 Chromium 方面的话题，让我受益匪浅。

最后要感谢我的太太、女儿和父母，在写作的大半年时间里给予了笔者很多支持。因为本书是在繁忙的工作之余利用琐碎的业余时间来完成的，所以，如果没有家人提供的良好环境，我是没有办法完成这本书的。特别是我的小女儿经常过来“光顾”和“巡视”我的写作，并给予一些特别的“惊喜”和“礼物”，让我在写作之余多了一份乐趣。

朱永盛

2014 年 2 月 1 日

目 录

第 1 章 浏览器和浏览器内核.....	1
1.1 浏览器	1
1.1.1 浏览器简介	1
1.1.2 浏览器特性	4
1.1.3 HTML	5
1.1.4 用户代理和浏览器行为	8
1.1.5 实践：浏览器用户代理	9
1.2 浏览器内核及特性	11
1.2.1 内核和主流内核	11
1.2.2 内核特征	12
1.3 WebKit 内核	15
1.3.1 WebKit 介绍	15
1.3.2 WebKit 和 WebKit2	16
1.3.3 Chromium 内核：Blink	18
1.4 本书结构	18
第 2 章 HTML 网页和结构	21
2.1 网页构成	21
2.1.1 基本元素和树状结构	21
2.1.2 HTML5 新特性	23

2.2 网页结构	25
2.2.1 框结构	25
2.2.2 层次结构	27
2.2.3 实践：理解网页结构	29
2.3 WebKit 的网页渲染过程	31
2.3.1 加载和渲染	31
2.3.2 WebKit 的渲染过程	32
2.3.3 实践：从网页到可视化结果	35
 第 3 章 WebKit 架构和模块	 39
3.1 WebKit 架构及模块	39
3.1.1 获取 WebKit	39
3.1.2 WebKit 架构	40
3.1.3 WebKit 源代码结构	43
3.2 基于 Blink 的 Chromium 浏览器结构	45
3.2.1 Chromium 浏览器的架构及模块	45
3.2.2 实践：从 Chromium 代码结构和运行状态理解现代浏览器	56
3.3 WebKit2	61
3.3.1 WebKit2 架构及模块	61
3.3.2 WebKit 和 WebKit2 嵌入式接口	62
3.3.3 比较 WebKit2 和 Chromium 的多进程模型以及接口	63
 第 4 章 资源加载和网络栈	 65
4.1 WebKit 资源加载机制	65
4.1.1 资源	65
4.1.2 资源缓存	67
4.1.3 资源加载器	68
4.1.4 过程	69
4.1.5 资源的生命周期	70
4.1.6 实践：资源的缓存	71
4.2 Chromium 多进程资源加载	74
4.2.1 多进程	74

4.2.2 工作方式和资源共享	76
4.3 网络栈	78
4.3.1 WebKit 的网络设施	78
4.3.2 Chromium 网络栈	78
4.3.3 磁盘本地缓存	84
4.3.4 Cookie 机制	88
4.3.5 安全机制	90
4.3.6 高性能网络栈	90
4.3.7 实践: Chromium 网络工具和信息	97
4.4 实践: 高效的资源使用策略	99
4.4.1 DNS 和 TCP 连接	99
4.4.2 资源的数量	99
4.4.3 资源的数据量	100
第 5 章 HTML 解释器和 DOM 模型	101
5.1 DOM 模型	101
5.1.1 DOM 标准	101
5.1.2 DOM 树	104
5.2 HTML 解释器	107
5.2.1 解释过程	107
5.2.2 词法分析	110
5.2.3 XSSAuditor 验证词语	111
5.2.4 词语到节点	111
5.2.5 节点到 DOM 树	113
5.2.6 网页基础设施	114
5.2.7 线程化的解释器	117
5.2.8 JavaScript 的执行	119
5.2.9 实践: 理解 DOM 树	120
5.3 DOM 的事件机制	121
5.3.1 事件的工作过程	122
5.3.2 WebKit 的事件处理机制	123
5.3.3 实践: 事件的传递机制	125

5.4 影子 (Shadow) DOM	127
5.4.1 什么是影子 DOM	127
5.4.2 WebKit 的支持	128
5.4.3 实践: 使用影子 DOM.....	129
 第 6 章 CSS 解释器和样式布局	131
6.1 CSS 基本功能	131
6.1.1 简介.....	131
6.1.2 样式规则.....	134
6.1.3 选择器.....	135
6.1.4 框模型.....	136
6.1.5 包含块 (Containing Block) 模型.....	139
6.1.6 CSS 样式属性.....	139
6.1.7 CSSOM (CSS Object Model)	140
6.1.8 实践: 理解 CSSOM 和选择器	141
6.2 CSS 解释器和规则匹配	143
6.2.1 样式的 WebKit 表示类.....	143
6.2.2 解释过程.....	146
6.2.3 样式规则匹配.....	148
6.2.4 实践: 样式匹配.....	149
6.2.5 JavaScript 设置样式	151
6.3 WebKit 布局	152
6.3.1 基础.....	152
6.3.2 布局计算.....	153
6.3.3 布局测试.....	155
 第 7 章 渲染基础	157
7.1 RenderObject 树	157
7.1.1 RenderObject 基础类.....	157
7.1.2 RenderObject 树.....	162
7.2 网页层次和 RenderLayer 树.....	163
7.2.1 层次和 RenderLayer 对象	163

7.2.2 构建 RenderLayer 树	165
7.3 渲染方式	167
7.3.1 绘图上下文 (GraphicsContext)	167
7.3.2 渲染方式	169
7.4 WebKit 软件渲染技术	172
7.4.1 软件渲染过程	172
7.4.2 Chromium 的多进程软件渲染技术	177
7.4.3 实践: 软件渲染过程	180
第 8 章 硬件加速机制	183
8.1 硬件加速基础	183
8.1.1 概念	183
8.1.2 WebKit 硬件加速设施	185
8.1.3 硬件渲染过程	189
8.1.4 3D 图形上下文	193
8.2 Chromium 的硬件加速机制	194
8.2.1 GraphicsLayer 的支持	194
8.2.2 框架	196
8.2.3 命令缓冲区	200
8.2.4 Chromium 合成器 (Chromium Compositor)	202
8.2.5 实践: 减少重绘	213
8.3 其他硬件加速模块	216
8.3.1 2D 图形的硬件加速机制	216
8.3.2 WebGL	223
8.3.3 CSS 3D 变形	228
8.3.4 其他	229
8.3.5 实践: Chromium 的支持	229
第 9 章 JavaScript 引擎	231
9.1 概述	231
9.1.1 JavaScript 语言	231
9.1.2 JavaScript 引擎	238

9.1.3	JavaScript 引擎和渲染引擎	241
9.2	V8 引擎	242
9.2.1	基础	242
9.2.2	工作原理	246
9.2.3	绑定和扩展	258
9.3	JavaScriptCore 引擎	259
9.3.1	原理	259
9.3.2	架构和模块	259
9.3.4	内存管理	265
9.3.5	绑定	266
9.3.6	比较 JavaScriptCore 和 V8	266
9.4	实践——高效的 JavaScript 代码	266
9.4.1	编程方式	266
9.4.2	例子	268
9.4.3	未来	271
第 10 章	插件和 JavaScript 扩展	273
10.1	NPAPI 插件	274
10.1.1	NPAPI 简介	274
10.1.2	WebKit 和 Chromium 的实现	275
10.2	Chromium PPAPI 插件	284
10.2.1	原理	284
10.2.2	结构和接口	285
10.2.3	工作过程	288
10.2.4	Native Client	294
10.3	JavaScript 引擎的扩展机制	297
10.3.1	混合编程	297
10.3.2	JavaScript 扩展机制	299
10.4	Chromium 扩展机制	303
10.4.1	原理	303
10.4.2	基本设施	306
10.4.3	消息传递机制	309

第 11 章 多媒体	311
11.1 HTML5 的多媒体支持	311
11.2 视频	313
11.2.1 HTML5 视频.....	313
11.2.2 WebKit 基础设施.....	315
11.2.3 Chromium 视频机制.....	317
11.2.4 字幕.....	328
11.2.5 视频扩展.....	330
11.3 音频	331
11.3.1 音频元素.....	331
11.3.2 Web Audio.....	334
11.3.3 MIDI 和 Web MIDI.....	336
11.3.4 Web Speech.....	337
11.4 WebRTC.....	339
11.4.1 历史.....	339
11.4.2 原理和规范.....	341
11.4.3 实践——一个 WebRTC 例子	342
11.4.4 WebKit 和 Chromium 的实现	345
第 12 章 安全机制	353
12.1 网页安全模型	353
12.1.1 安全模型基础.....	353
12.1.2 WebKit 的实现	363
12.2 沙箱模型	366
12.2.1 原理.....	366
12.2.2 实现机制.....	367
第 13 章 移动 WebKit	373
13.1 触控和手势事件	373
13.1.1 HTML5 规范	373
13.1.2 工作原理.....	377
13.1.3 启示和实践.....	381

13.2 移动化用户界面	382
13.3 其他机制	384
13.3.1 新渲染机制	384
13.3.2 其他机制	387
第 14 章 调试机制	389
14.1 Web Inspector	389
14.1.1 基本原理	389
14.1.2 协议	391
14.1.3 WebKit 内部机制	395
14.1.4 Chromium 开发者工具	398
14.1.5 远程调试	400
14.1.6 Chromium Tracing 机制	402
14.2 实践——基础和性能调试	404
14.2.1 基础调试	404
14.2.2 性能调试	408
第 15 章 Web 前端的未来	411
15.1 趋势	411
15.2 嵌入式应用模式	414
15.2.1 嵌入式模式	414
15.2.2 CEF	414
15.2.3 Android WebView	417
15.3 Web 应用和 Web 运行环境	419
15.3.1 Web 应用	419
15.3.2 Web 运行环境	421
15.4 Cordova 项目	423
15.5 Crosswalk 项目	425
15.6 Chromium OS 和 Chrome 的 Web 应用	429
15.6.1 基本原理	429
15.6.2 其他 Web 操作系统	431
参考资料	435

1

第 1 章 浏览器和浏览器内核

浏览器是目前用户使用范围最广、使用时间最长的应用程序之一，浏览器的发展也经历了一段坎坷的过程。伴随着浏览器发展的是浏览器内核，它是浏览器中最核心的功能部件，本章作为本书的开始，在基础性介绍了浏览器和浏览器内核等概念之后引入了 WebKit 内核的特征分析和框架阐述。

1.1 浏览器

1.1.1 浏览器简介

互联网的革命浪潮带动了众多技术的快速发展，其中，网络浏览器（之后将简称为浏览器）作为互联网最重要的终端接口之一在短短的二十多年时间里日新月异，特别是在进入 21 世纪后，越来越多的功能被加入到浏览器中来。在 W3C 等标准组

织的积极推动下逐步成型的 HTML5 技术，更是成为了浏览器发展的火箭推进器。

提到浏览器，不得不提的重量级人物是 Berners-Lee。Berners-Lee 是 W3C 组织的理事，他在 80 年代后期 90 年代初期发明了世界上第一个浏览器 WorldWideWeb(后改名为 Nexus)，并在 1991 年公布了源代码。它支持早期的 HTML 标记语言，当然，它的功能也很简单，只是支持文本、简单的样式表、电影、声音和图片等。但是，在当时的情况下，它是仅有的能够可视化网络内容的浏览器。

第二个不得不提的重量级人物是 Marc Andreessen。在 1993 年，真正有影响力的浏览器 Mosaic 诞生，它是由 Marc Andreessen 领导的团队开发，这就是后来鼎鼎大名的网景(Netscape)浏览器。同样地，在最开始的时候，它所支持的功能也有限，只能显示简单的静态 HTML 元素，没有 JavaScript，没有 CSS，更没有目前 HTML5 各种丰富的功能。不过，网景浏览器还是大受欢迎，获得世界范围内的成功，之后发展迅速，在其顶峰时期，占据了绝大多数的市场份额。

事情的转变源于 1995 年。受 Mosaic 浏览器的深刻影响，微软推出了闻名世界的 Internet Explorer（以下简称为 IE）浏览器，自此第一次浏览器大战正式打响。IE 受益于 Windows 操作系统，获得了空前的成功，其逐渐取代了网景浏览器的领导地位，一直到网景浏览器的消亡，至此，第一次浏览器大战结束。

处于低谷的网景公司在 1998 年成立了 Mozilla 基金会，开始凤凰涅槃。在该基金会的推动下，网景公司主导开发了著名的开源项目火狐浏览器（也就是 Firefox，后面使用该名称），在 2004 年发布了 1.0 版本，拉开了第二次浏览器大战的序幕，这次大战影响深远。受益于 IE 浏览器发展较为缓慢，Firefox 浏览器自推出以来就深受大家的喜爱，其功能丰富，扩展众多，因此市场份额一直在上升。

然而，第二次浏览器大战并没有结束，就在 Firefox 浏览器发布 1.0 版本的前一年，也就是 2003 年，苹果发布了 Safari 浏览器，并在 2005 年释放了浏览器中一种非常重要部件的源代码，发起了一个新的开源项目 WebKit（它是 Safari 浏览器的内核，