禪

همکاری برای طراحی WebAssembly

اتحاد گوگل، مایکروسافت و موزیلا برای افزایش سرعت مرورگرهای وب

مهندسان شرکتهای گوگل، مایکروسافت، اپل و موزیلا گرد هم آمدند تا WebAssembly که به اختصار wasm خوانده میشود را طراحی کنند . یک بایتکد (bytecode) ویژه که در آینده مورد استفاده مرورگرها قرار خواهد گرفت و ادعا میکند کارایی را به میزان 20 برابر سریعتر خواهد کرد .

ByteCode چیست؟

ByteCode که بهنام کد قابل حمل (Portable code) یا بهاختصار p-code نیز نامیده میشود، شکلی از مجموعه دستورالعملها (Instruction set) است که برای اجرای مؤثر توسط یک مفسر نرمافزار طراحی شدهاند. برعکس کدهای منبع (Source code) که به صورت قابل فهم انسانی هستند، بایتکدها از کدهای عددی فشرده، ثابتها و ارجاعات (معمولا آدرسهای عددی) ساخته شدهاند که نتیجه تجزیه و تحلیلهای معنایی اشیایی نظیر نوعها، دامنه و اشیاء تودرتو و داخلی یک برنامه را کدگذاری یا به عبارت دقیقتر رمزنگاری (Encode) میکنند . از این رو کارایی خیلی بهتری نسبت به ترجمه مستقیم کد اصلی خواهند داشت. یک برنامه بایتکد ممکن است فرآیند تجزیه و اجرای مستقیم دستورالعملها را در یک زمان انجام دهد . این نوع از مفسران بایتکدها بسیار قابل حمل هستند . بعضی سیستمها از آنها به نام مترجمان پویا یا کامپایلرهای just in time (JIT) نام میبرند . از جمله بایتکدهای معروفی که امروزه مورد استفاده قرار میگیرند، میتوان به CLISP، Byte Code Engineering Library، ActionScript،

Common Intermediate Language که توسط محیط زمان اجرای CLR اجرا شده و توسط زبانهای برنامهنویسی داتنت همچون سیشارپ مورد استفاده قرار میگیرند، Java bytecode، Emacs و.... اشاره کرد .

جاوااسکریپت دوستداشتنی

برای سالهای متمادی موتورهای جاوااسکریپت کانون توجه توسعهدهندگان مرورگرها قرار داشتند . همین موضوع باعث شده بود تا محصولات سازندگان نه تنها روند رو به رشدی داشته باشد بلکه عملکرد و بهرهوری آنها به میزان قابل توجهی بهبود یابد . javascript engine) یک ماشین مجازی است که برای ترجمه و اجرای دستورات جاوااسکریپت مورد استفاده قرار میگیرد . البته موتورهای جاوااسکریپت کارکردهای مختلفی دارند و به طور گسترده توسط مرورگرهای وب مورد استفاده قرار میگیرند .) کامپایل پویا که از آن بهنام کامپایل درجا نیز یاد میشود، برای تبدیل کدهای جاوااسکریپت به دستورالعملهایی مورد استفاده قرار میگیرد که به طور مستقیم روی پردازشگر اجرا شده و به لحاظ سرعت دستاوردهای مهمی را بههمراه میآورد . با وجود این پیشرفتها هنوز یک پرسش وجود دارد : "چرا جاوا اسکریپت؟ ". البته بهکارگیری جاوااسکریپت هزینههایی را نیز تحمیل میکند . مرورگرها باید کدهای زبانی که بر پایه متنهای قابل فهم انسانی و نه ماشینی است را دریافت کرده، آنها را خوانده و ترجمه کنند . حتی ساختار خود جاوااسکریپت بهگونهای است که دارای تعدادی ویژگی نه چندان مطلوب است که همین موضوع بر کارایی آن تأثیر نامطلوب میگذارد . به طور مثال، روشی که در آن یک متغیر در شرایط مختلف میتواند انواع متنوعی از دادهها همچون اعداد، رشتهها یا بخشی از HTML ترجمه شده را در خود جای دهد، اما یک کامپایلر JIT ممکن است توانایی ترجمه بهینهسازی شده آنرا بهگونهای که لازم است نداشته باشد یا توانایی ویرایش رفتار اشیاء از پیش ساخته شده شبیه آرایهها که ممکن است دردسرهایی را بهوجود آورد، نمونهای از این معایب بهشمار میرود . اما جاوااسکریپت مزیتهای کاملا مشخصی نیز دارد . برنامههای جاوااسکریپت در محیط sandbox قرار دارند؛ به این معنی که به غیر از باگهای مرورگرها، برنامههای جاوااسکریپت نمیتوانند فراتر از محدوده مرورگر برای دسترسی به اطلاعات حساس یا نصب نرمافزارهای مخرب اقدام کنند . جاوااسکریپت همچنین مستقل از پردازشگر است، در نتیجه اسکریپتها به همان خوبی که روی یک کامپیوتر شخصی با معماری x86 اجرا میشوند روی گوشیهای هوشمند مجهز به پردازنده آرم نیز اجرا میشوند . همین نکات مثبت باعث میشود تا مزیتهای جاوااسکریپت بدون آن که جنبههای منفی آن به چشم آیند مورد توجه قرار گیرند .

تقاضا برای بهکارگیری بایتکدها آغاز میشود

به این ترتیب فشارهایی برای استفاده از سیستمهای بایتکد در مرورگرها بهوجود آمد . بر همین اساس مایکروسافت و سان (در حال حاضر اوراکل) این موضوع را با داتنت و جاوا حل کردند اما این سیستمها به جای آنکه بر موتور رندر مرورگرها و به صورت یکپارچه با جاوااسکریپت استوار باشند، بر افزونهها متکی بودند . برنامههای جاوا اسکریپت به طور مستقیم توانایی دستکاری اشیاء HTML را دارند اما در عوض افزونهها دنیای خاص خود را دارند . شرکتهای دیگری همچون گوگل سعی کردند مجموعهای از سیستمها را برای مرورگرها توسعه داده و اتکا بر جاوااسکریپت را کم کنند . Native Clinet که برنامههای بر پایه معماری x86 یا آرم را در یک محیط ایمن سندباکس اجرا میکرد یا Portable Native Client (PNaCl)

نیز همین کار را انجام میداد اما از نوعی بایتکد به جای استفاده از کد آرم یا x86 استفاده میکرد . در حالیکه گوگل سعی کرد کمپینی را برای استفاده از ابداعات خود به راه اندازد، اما سازندگان مرورگرها از پذیرفتن آنها طفره رفتند . در طول این سالها راهکارهای مختلفی ارائه شدند و همچون نمونههایی که به آنها اشاره کردیم توسط شرکتهای بزرگ مطرح شدند اما یک اجماع واحد روی آنها به وجود نیامد . مردم خواستار یک بایتکد ویژه برای مرورگرها بودند که مزایایی که در گذشته وجود نداشت را در بر گرفته و ایرادات قبلی را نداشته باشد .

WebAssembly به میدان وارد میشود

وباسمبلی، پروژهای جدید بوده که کار روی آن توسط موزیلا، مایکروسافت، گوگل و اپل در حال انجام است و هدف از آن تولید یک بایتکد برای وب است که به درخواستها واکنش نشان دهد .

Microsoft
mozilla
Google
GitHub
Explore
Features
Enterprise
Blog
Sign up
Sign in
WebAssembly
Development of WebAssembly and associated infrastructure
People
Filters
v8-native-prototype
design
WebAssembly Design Documents
js-astcompressor-prototype
experiments
polyfill-prototype-1
Experimental WebAssembly polyfill library and tools

WebAssembly که به اختصار wasm نامیده میشود با هدف ارائه بایتکدی قابل حمل که بر روند دانلود و بارگذاری تأثیرگذار بوده و کارآمدتر از جاوا اسکریپت یا asm.js باشد، در حال طراحی است. WebAssembly پروژهای است که اکنون روی گیتهاب میزبانی میشود و برای ساخت یک بایتکد ماشینی که توانایی خواندن مجموعهای از دستورالعملها را برای مرورگرها راحت کرده بهطوری که توانایی بارگذاری زبانهای سطح بالا را داشته باشند باعث به وجود آوردن کارایی محسوسی برای مرورگرهای دسکتاپ و موبایل خواهد شد به طوریکه امکان تجزیه کامل کدهای منبع یک صفحه وب یا یک برنامه را فراهم میکند .

JavaScript source
WebAssembly
Bytecode
Machine code

مرورگرها در حال حاضر از جاوااسکریپت برای تفسیر کدها و فعالسازی قابلیتهایی نظیر محتوای پویا و فرمها روی سایتها استفاده میکنند . هر چند راهکارهایی برای بهبود زمان بارگذاری از طریق asm.js انجام شده است، اما سیستمهای بایتکد محور شبیه به داتنت سریعتر هستند .

ساختار فعلی وباسمبلی چگونه است؟

طراحی وباسمبلی بهگونه ای خواهد بود که هم شامل نمادهای دودویی که توسط کامپایلرها تولید میشوند و هم شامل متن متناظر که برای نمایش در دیباگرها و برای محیطهای توسعه مناسب است خواهد بود . نمونههای اولیه که تاکنون ارائه شدهاند از ویژگیهای هیجانبرانگیز و کارآمد خبر میدهند بهطوری که آمارها نشان از 20 برابر سریعتر بودن تجزیهها نسبت به asm.js حکایت دارند . مهندسانی که در پشت صحنه wasm قرار دارند فراموش نکردهاند که جاوا اسکریپت در همه جا پشتیبانی میشود و wasm در حال حاضر در هیچ کجا از جاوااسکریپت پشتیبانی نمیکند . آنها در نظر دارند تا این خلاء را از طریق polyfill یک اسکریپت جاوااسکریپت که قادر به تبدیل wasm به asm.js برای مرورگرهایی که به طور محلی از wasm پشتیبانی نمیکنند پیادهسازی کنند . البته wasm هنوز در مرحله توسعه قرار دارد . هنوز هیچ استانداردی در بدنه آن به کار نرفته است و تنها یک گروه غیر رسمی در حال کار روی آن هستند . مشخصات هنوز کامل نشدهاند و طراحی سطح بالای آن هنوز در مرحله تصمیمگیری قرار دارد . اما هر چهار سازنده بزرگ موتور مرورگرها با یکدیگر در حال کار روی آن هستند .

WEBASSEMBLY
A Google, Microsoft, Apple And Mozilla Initiative
Let's Build a Faster Web

آیندهای روشن

آینده wasm باید درخشان باشد و منتقدان جاوااسکریپت که برای مدتهای طولانی از جاوااسکریپت

ناراضی بوده و شکایت میکردند اکنون به آرزوی خود برای داشتن یک بایتکد ویژه خواهند رسید . پیشنهاد این فناوری در قالب یک استاندارد روزگاری در همه مرورگرها پیادهسازی خواهد شد . تا زمانیکه WebAssembly به طور گسترده در دسترس قرار گیرد، ائتلاف توسعهدهندگان در حال برنامهریزی برای پر کردن این شکاف با استفاده از JS script هستند که توانایی تبدیل wasm به asm.js را در اختیار مرورگرهایی که هنوز از فرمت جدید پشتیبانی نمیکنند قرار WebAssembly.□□□ هنوز تا رسیدن به مرحله نهایی زمان زیادی خواهد داشت : نه مشخصات آن و نه طراحی سطح بالای آن هنوز به پایان نرسیدهاند . با این حال، با توجه به اینکه توسعهدهندگان بزرگ در پشت این پروژه قرار دارند باید در آیندهای نزدیک شاهد افقهای روشنی در این زمینه باشیم .

نویسنده :
حمیدرضا_تائبی
دسته بندی:
شاهراه اطلاعات
تاریخ انتشار :
23:24 - 07/04/1394
برچسب:
جاوااسکریپت - wasm - WebAssembly - بایت_کد - - javascript engine - just in time (JIT) - bytecode
polyfill - مرورگر وب

□□□□□ منبع: https://www.shabakeh-mag.com/information-feature/939