



黑猿大叔  
40 篇文章

【译文】为什么说Rust是机器人技术的未来

关注作者

写文章

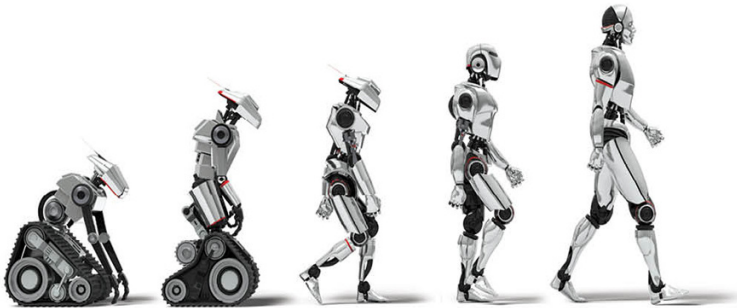
目录

专栏首页 TensorFlow从0到N 【译文】为什么说Rust是机器人技术的未来

【译文】为什么说Rust是机器人技术的未来

2020-06-09 阅读 73

原文：Why Rust is the future of robotics  
作者：Pierre Rouanet



Robot Evolution

Rust是一门比较新的编程语言，在2006年由Graydon Hoare启动，2010年由Mozilla正式发布。速度极快的火狐量子浏览器就是用Rust开发的。过去几年里，Rust凭借着许多超棒的特性，产生了很大的吸引力，尤其对机器人技术。

Rust吸收了像C和C++这些低级语言的特性——运行速度极快且占用内存少，同时具有很多高级语言的特性保障。它可以避免段错误、保证线程安全。需要特别指出的是Rust在保障所有这些安全特性和抽象特性的同时并不产生额外的开销。它的座右铭：

*Rust：快，可靠，高产——一石三鸟*

基于上述，我们相信Rust是机器人技术的未来，它使开发者无所畏惧，去创建和执行标准，去建立并依靠一个强大的社区。我们接下来会更详细的解释，在此之前先回顾下当今机器人技术中的嵌入式软件困局。

嵌入式编程的困局

在嵌入式软件中存在两大难题：

很难写出安全的代码。C和C++的内存使用不安全，易发人为错误。很多的bug源于缓冲区溢出，数据损坏，读写未分配的内存。检测和调试变得非常困难，很多攻击和病毒便趁虚而入。你的项目越复杂，就会越难维护。换句话说，嵌入式程序员最担心：添新功能，毁所有。

在嵌入式软件中几乎无法实现并发，所以很难完成从玩具示例到量产的跨越。竞争条件，死锁，数据损坏导致的bug调试起来本来就复杂。对于嵌入式系统来说，开发环境相比基于OS的世界就更没优势了。调试硬件中断甚至会让信心满满的嵌入式开发者感到恐惧。

你可以在这里找到更多关于为什么嵌入式软件如此艰难。

除了这些技术层面的问题，尤其在机器人领域，还缺少标准化和可直接复用的代码。当然，Internet上总能找到你想要的示例代码。但是通常不能拿来直接用。这导致开发人

嵌入式编程的困局

无所畏惧

作者介绍  
说说机器人技术  
一流的社区



黑猿大叔

果藤金融图像算法工程师

关注

专栏

文章	阅读量	获赞	作者排名
40	19.5K	245	2099

精选专题

云+社区×知乎「AI与传统行...  
AI 具有什么能力？能给传统行业带来哪些变革与发展？

活动推荐

腾讯云自媒体分享计划

入驻云加社区，共享百万资源包。

立即入驻

邀请作者加入自媒体计划

每月最高可拿1800元无门槛代金券。

了解更多



黑狼大叔  
40 篇文章

【译文】为什么说Rust是机器人技术的未来

关注作者

写文章

分享

Rust可以解决所有这些问题，甚至更多。

无所畏惧

编译即安全。再重复一遍。编译即安全。

保证。对于嵌入式开发来说，这是极大的缓解。Rust是如何做到的？本篇不打算深入探讨，让我简单说一下。

Rust中的一个重要原则是所有权。所有权跟踪并确保每个变量在同一时刻只能有一个所有者，如果所有者超出范围，该值就被释放。这个原则非常强悍，它一次性解决了两个重要问题：内存安全和无痛并发，通常来自在不该访问数据的时候进行访问。

**内存安全**——Rust不允许空指针和悬挂指针。也没有垃圾收集器来确保可预测的代码行为。

**无痛并发**——基于借用概念，Rust能跟踪到数据竞争风险的存在，从而不进行编译。它提供了高级的机制把这些条件封装在互斥锁，利于你继续。

这已经有点意思了，但是要花费多少呢？好吧，几乎为零。

**零成本抽象**——抽象是计算机科学中的一个发明。John V Guttag给了一个很好的定义：“抽象的本质是保留和上下文相关的信息，忽略与上下文不相关的信息”。这就是之所以像Python这类高级语言产生的原因。但是Python附带了很多开销处理，都难以跟踪，比如垃圾回收，运行期指针的检查等等。但是嵌入式系统经常处理关键应用，最差情况执行时间是必须的。想想数控制动器。你猜对了，Rust允许抽象，但没有开销。大部分的检查都在编译期进行，所以在运行期没有额外的计算。

Rust还提供：

现代化的语法

精确的错误信息

无痛打包和依赖管理——Rust搭配有最好包管理器，叫Cargo。对于嵌入式编程，有Xargo。

对于嵌入式系统，还开发了专用工具：

**周边控制**——有个叫做svd2rust的东东，它可以直接从SVC描述为每个周边控制器自动生成Rust API。Rust很聪明，它可以在编译时强制您使用这些外围设备。如果尝试写入只读寄存器或读取只写寄存器，则无法编译。你也不能将无效的位模式写入寄存器。SVD会定义一个有效值的范围，Rust不会让你超出范围。

**资源冲突预防**——在下个版本，他们将引入单例，让Rust知道何时代码想使用一个正在使用的外围设备，比如一个定时器。这是在嵌入式系统中常见的问题根源，即多个设备想要使用有限的资源。一个经典案例就是Arduino Uno开发板，上面只有很少的定时器。所以当你使用Servo库时，你不能在引脚9和10上使用PWM。但是代码可以编译，加载到开发板上，然后产生各种奇怪的行为，且难以调适，并最终导致严重的失败。这使得许许多多的Arduino用户感到困惑。有了单例，Rust就可以在编译期告诉你，定时器是否已经在使用了，这样就可以避免掉很多让用户头痛的问题，而且没有任何的运行期开销。

不要担心，Rust同时兼容C和C++。

如果你以前从事过嵌入式编程，你一定在这些问题上花了不少精力。Rust不仅会让你对自己的代码有信心，而且对你想重用的其他人的代码也有信心。

说说机器人技术

对于嵌入式系统的代码共享和重用，有两个主要的挫折点：

第一，每个人的工作都可以进行分类，比如读取一个模拟值，记录一个编码器的位置，更改马达的速度或者LED的颜色。但是，没有人使用相同的接口，一个马达的两个实现总是使用不同的函数名和API来做同一件事情。这使得写一个适用于两个马达的通用马达控制器变得很困难。

第二，源于第一点，就是我们希望重用别人的代码时能充满信心。我们希望确保不会引入新的bug，因为我们使用代码的场景是最初的开发者没有测试过的。

目录

嵌入式编程的困局

无所畏惧

说说机器人技术

一流的社区

1

0

分享

不安全的情况)。还有很多其他方面, 这里我强调两个重要功能:

**Traits**——和面向对象编程类似, 类和集成? Rust Traits用于相似类型的抽象, 但是与继承不同, 它使用**组合**原则。通过组合, 能定义一个可以安全混合和匹配的属性集合。所有这些的计算开销为零。

**类型推断**——简单的说, 如果一个函数要求输入角度而你输入了弧度, Rust能自动的检测到需要进行类型转换。而且如果定义了类型转换函数, 它会自动使用它把弧度转换为角度。在机器人技术领域, 我们经常会做一些数学运算, 而使用错误的单位常常导致糟糕的意外, 有些花费数百万美元。幸运的是Rust回来了。你可以在这里阅读更多关于类型推断的信息。如果你对组合有兴趣, 可以看下这个视频。

如果我们希望机器人技术能按预期快速传播, 标准化就是非常关键的。我们对Rust能够进行大规模的共享充满信心。

### 一流的社区

在2016和2017年的Stack Overflow开发者调查中, Rust获得了“最受欢迎开发语言”第一名。这真的不稀奇。Rust社区有热情, 有组织, 很活跃。

**热情**——去年聚焦的一个重点就是降低初学者的学习曲线, 为各个级别提供指导, 在编辑-编译-调试循环过程中提升用户体验。

**有组织**——Rust有一个完善的文档系统, 它覆盖了所有的标准库和错误码。Rust社区有一套行为守则而且每年他们会总结社区成就并在所有社区的帮助下产生一个新的路线图。他们建立了一个完善的评论请求流程RFC征求社区对于新发展和方向的反馈。

**活跃**——我们在4个月前开始使用Rust开发库, 为的是机器人产品开发能够变得简单, Rust社区的活力实在帮了我们快速起步。就在这周, 在Philipp Oppermann的可用于no\_std系统的简单分配器linked-list-allocator项目我们开启了一个拉取请求。不到两小时我们的拉取请求就被集成了。还不止一次。

这个社区在嵌入式方面非常的积极并正在成长, 有很多积极的Embedded in Rust开发者和博客作者。雄心勃勃的项目正在涌现, 在实时框架和安全的嵌入式操作系统, 有望迎来光明的未来。甚至Android也正在转向Rust。

无需多说一个有组织和有纪律的社区, 使你感受到热情、被倾听和受支持, 对我们在机器人领域使用和开发Rust, 是一个强大的动力。

我希望使你相信, Rust在嵌入式编程尤其是机器人技术方面具有巨大的潜力。但是Rust嵌入式社区仍然很小而且很多东西都有缺失。不要害怕, 创造, 构建, 重用标准并享受社区的乐趣!

刚开始, 你可能会感觉到这是你和Rust编译器之间的斗争: 写代码→ 尝试编译→ 编译失败→ 写代码的方式使其不安全→ 羞辱编译器→ 否则继续尝试→ 再次编译→ 可以用了→ 咕咚“好吧这样的确更好些”。)但是这个旅程中Rust会使你变成更好的开发者, 并极大的帮助你写出安全和高效的代码!

我们将会持续写关于Rust和机器人技术的博客, 解释在Rust in Embedded中我们的工具链, 工作流, 专用库, 编程技巧, 以及我们如何构建机器人。我们已经在Github上开源了我们所有的工作: <https://github.com/pollen-robotics>。

我是一名软件工程师/博士, Pollen Robotics的联合创始人。

本文参与[腾讯云自媒体分享计划](#), 欢迎正在阅读的你也加入, 一起分享。

举报

点赞 1

分享

0 条评论

我来说两句

目录

- 嵌入式编程的困局
- 无所畏惧
- 说说机器人技术
- 一流的社区



黑猿大叔  
40 篇文章

【译文】为什么说Rust是机器人技术的未来

关注作者

写文章

目录

- 嵌入式编程的困局
- 无所畏惧
- 说说机器人技术
- 一流的社区

相关文章

释放堆内存，Rust是怎么做的？所有权！

这个概念是支撑Rust在编译期做内存安全检查的核心机制，也正是因为这个特性，我们认为Rust是内存安全的底层语...

黑猿大叔

讨厌算法的程序员 4 - 时间复杂度

增长量级？函数的增长量级 上一篇算法分析基础中，我们分析了插入排序，知道了其最好情况下的运行时间为 $T(n) = a...$

黑猿大叔

TensorFlow从0到1 - 10 - NN基本功：反向传...

上一篇 9 “驱魔”之反向传播大法引出了反向传播算法——神经网络的引擎，并在最后窥探了它的全貌。本篇将详细的...

黑猿大叔

【Rust日报】2019-07-31：Debian Buster ...

两个指针a和b有可能互为别名（俩指针指向同一数据），所以clang在编译该代码的时候，会在寄存器中存储两次...

MikeLoveRust

一步步靠近：Rust入门小百科

从2015年Rust 发布1.0 版本以来，Rust 语言已经被广泛应用于各大公司及诸多领域。每一年，Rust 社区都会聚集在一...

用户1682855

在 RustCon Asia 开启之前，聊聊 Rust 中国社区那些事

2012 年 1 月 24 日，在中国最大的问答社区「知乎」，名为“题叶”的网友，创建了「Rust（编程语言）」话题，那时候这门语言还无人问津。2013 年 ...

PingCAP

编程语言中的变革者 | 敢于打造理想世界的 Rust

在历史的早期，程序员们写代码，都是直接机器码编程，就是纸带机，大家都在那戳一个个小孔来编程。

张汉东

Rust 编译模型之殇

本文转载自 PingCap 公众号。原文地址：

分享



黑猿大叔  
40 篇文章

【译文】为什么说Rust是机器人技术的未来

关注作者

写文章

【深度知识】Rust语言入门、关键技术与实战经验

编者按：高可用架构分享及传播在架构领域具有典型意义的文章，本文由唐刘在高可用架构群分享。转载请注明来自高可用架构公众号「 ArchNotes 」。

辉哥

我用 Rust 重写了博客，谈谈我眼中的 Rust.

我已经学习 Rust 有挺长的一段时间了，一直想用 Rust 来写点东西，但是缺乏好的想法；有些想法自我感觉良好，但...

春哥大魔王

更多文章

目录

- 嵌入式编程的困局
- 无所畏惧
- 说说机器人技术
- 一流的社区

分享

社区

- 专栏文章
- 互动问答
- 技术沙龙
- 技术快讯
- 团队主页
- 开发者手册
- 智能钛AI

活动

- 原创分享计划
- 自媒体分享计划
- 邀请作者入驻
- 自荐上首页
- 在线直播
- 生态合作计划

资源

- 腾讯云大学
- 技术周刊
- 社区标签
- 开发者实验室

关于

- 视频介绍
- 社区规范
- 免责声明
- 联系我们

云+社区



扫码关注云+社区  
领取腾讯云代金券

热门产品

域名注册	云服务器	区块链服务	消息队列	网络加速	云数据库	域名解析
云存储	视频直播					
人脸识别	腾讯会议	企业云	CDN 加速	视频通话	图像分析	MySQL 数据库
SSL 证书	语音识别					
数据安全	负载均衡	短信	文字识别	云点播	商标注册	小程序开发
网站监控	数据迁移					