Atitit FFI(外部函数接口[Java Foreign Function Interface](http://openjdk.java.net/jeps/191" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)

目录

[1. FFI(外部函数接口Java Foreign Function Interface（FFI） 1](#_Toc25853)

[1.1. Java的FFI(外部函数接口 jna jnr技术 1](#_Toc2467)

[1.2. 本节将分别为大家讲解 Go语言是如何与 C/C++ 进行交互的。 2](#_Toc18395)

[1.3. 与 C语言进行交互 2](#_Toc29861)

# FFI(外部函数接口[Java Foreign Function Interface](http://openjdk.java.net/jeps/191" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)（FFI）

## Java的FFI(外部函数接口 jna jnr技术

因此JNI创建本地函数的方式并不简单，于是产生了像[Java Native Access](https://github.com/twall/jna" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)（JNA）和[Java Native Runtime](https://github.com/jnr/jnr-ffi" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)（JNR）这样的库。JNA和JNR都是基于JNI创建的，而JEP 191定义的[Java Foreign Function Interface](http://openjdk.java.net/jeps/191" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)（FFI）可能会基于JNR。使用FFI API而不是JNI绑定本地代码和内存将成为开发人员更喜欢的方式。

FFI API将提供下列特性：

* 一个描述本地库调用和本地内存结构的元数据系统。
* 发现和加载本地库的机制。
* 基于元数据将库/函数或内存结构绑定到Java端点的机制。
* 用于Java数据类型和本地数据类型之间编组和解组的代码。

对Java FFI的需求已经产生了JNA和JNR库。JNA库应用更广泛（具体使用参见“[JNI的替代者—使用JNA访问Java外部函数接口](http://blog.csdn.net/lanxuezaipiao/article/details/22619835" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)”）。JNR库更全面，因为它实现了不同层次的抽象，提供了函数和内存元数据，对库和函数绑定进行了抽象。JNR已经在JRuby项目中大量使用，它可能会成为JEP 191的基础。

上面段落来自JEP 191的描述（由参考文献（1）翻译），由此可见虽然JNA使用广泛，但JNR可能更渐趋势，也许在不久的将来JNR-FFI（jffi）就会内建在JDK中与JNI一样成为Java访问外部函数的标准接口。因此，学习使用JNR是非常有必要的。

[JNR-FFI](https://github.com/jnr/jnr-ffi" \t "https://www.cnblogs.com/lanxuezaipiao/p/_blank)项目也托管自Github，其使用方法与JNA差不多，不过JNR并没有给出相应的jar包，需要我们自己打包使用。

## 本节将分别为大家讲解 [Go语言](http://c.biancheng.net/golang/" \t "http://c.biancheng.net/view/_blank)是如何与 C/[C++](http://c.biancheng.net/cplus/" \t "http://c.biancheng.net/view/_blank) 进行交互的。

## 与 [C语言](http://c.biancheng.net/c/" \t "http://c.biancheng.net/view/_blank)进行交互

工具 cgo 提供了对 FFI（外部函数接口）的支持，能够使用 Go语言代码安全地调用 C语言库，可以访问 cgo 文档主页：[https://golang.google.cn/cmd/cgo/](https://golang.google.cn/cmd/cgo/" \t "http://c.biancheng.net/view/_blank)。cgo 会替代 Go 编译器来产生可以组合在同一个包中的 Go 和 C 代码。  
  
在实际开发中一般使用 cgo 创建单独的 C 代码包。如果想要在 Go 程序中使用 cgo，则必须在单独的一行使用 import "C" 来导入，一般来说你可能还需要 import "unsafe" 。  
  
然后，可以在 import "C" 之前使用注释（单行或多行注释均可）的形式导入 C语言库（甚至有效的 C语言代码），它们之间没有空行，例如：

1. // #include <stdio.h>
2. // #include <stdlib.h>
3. import "C"

名称 "C" 并不属于标准库的一部分，这只是 cgo 集成的一个特殊名称用于引用 C 的命名空间。在这个命名空间里所包含的 C 类型都可以被使用，例如 C.uint 、C.long 等等，还有 libc 中的函数 C.random() 等也可以被调用。

JNI的又一替代者—使用JNR访问Java外部函数接口(jnr-ffi) - Alexia(minmin) - 博客园.mhtml

Go语言与C\_C++进行交互.mhtml