

## C语言：

C语言已经有40多年的历史了，是为了编写 UNIX 操作系统而被发明的，到目前为止仍然是最广泛使用的系统程序设计语言。Unix/Linux的内核以及其中的很多软件都是用C写的，Mysql和Apache也是用C写的，很多语言的编译器或解释器也都是用C写的，比如CPython，GCC。

### 为什么要使用 C？

C 语言最初是用于系统开发工作，特别是组成操作系统的程序。由于 C 语言所产生的代码运行速度与汇编语言编写的代码运行速度几乎一样，所以采用 C 语言作为系统开发语言。

### C语言的缺陷？

模块化能力和对include头文件的依赖

灵活的指针也是危险的指针-指针运算

不严格的语法限制和类型约束

蹩脚的字符串类型，没有bool类型

古怪的语法，易读性差

## Go语言：

Go是从2007年末由Robert Griesemer, Rob Pike, Ken Thompson主持开发，后来还加入了Ian Lance Taylor, Russ Cox等人，并最终于2009年11月开源，在2012年早些时候发布了Go 1稳定版本。现在Go的开发已经是完全开放的，并且拥有一个活跃的社区。

Go语言可以在不损失应用程序性能的情况下降低代码的复杂性。Go语言专门针对多处理器系统应用程序的编程进行了优化，使用Go编译的程序可以媲美C或C++代码的速度，而且更加安全、支持并行进程。

Docker以及Terraform是用Go语言写的。

### 语言特点：

简洁、快速、安全

并行、有趣、开源

内存管理、数组安全、编译迅速

### Go语言的简洁：

Go语言有25个非数据类型关键字；C语言有26个非数据类型关键字；Java有42个非数据类型关键字。

Go有三种方式的循环，只使用for一个关键字来实现；Java和C都有三种循环模式，通过for，do，while三个关键字实现；

Go中与面向对象相关的关键字只有2个（interface和func）；Java中与面向对象相关的关键字有14个

（abstract，instanceof，new，extends，final，interface，private，this，class，implements，protected，static，public，super）。

Go中的关键字func既能用来声明函数，也能用来声明方法。

## Go与C的对比：

以下从数据、计算、模块化、编程模型四个方面来比较C与Go。

## 数据：

**C语言：**

**Go语言：**

## 计算：

**C语言：**

C语言中高级别的计算就是函数；而低级别的计算就是语句和表达式。

**Go语言：**

Go语言中高级别的计算主要有函数、方法和接口；低级别的计算依然是语句和表达式，但Go语言通过增加新的语句和表达式使得Go语言的代码更加简洁易读了。

**函数、方法、接口：**

函数是Go语言中计算的最基本封装单位。

方法通过将函数和数据结合起来。

接口则是一组方法的集合。

**语句、表达式：**

**新语句：**

go：用来支持并发编程。

defer：实现类似Java中的finally的机制，以便能更安全的释放资源，如关闭文件等。

switch：switch有了match等新功能。

for：for有多种形式。

select：

**新表达式：**

<-：用来从Channel中读取数据或者向Channel中写入数据。

range：用来遍历array，slice，map，channel。

## 模块化：

**C语言：**

**源代码：**

.c、.h

**模块化：**

C语言在语言级别没有模块化，只是在编译环节有模块化。

**作用域：**

C语言只有全局和局部（函数或块内）两种作用域，模块化和隔离性比较差。

## Go语言：

### 源代码：

.go

### 模块化：

package

### 作用域：

Go语言通过package在语言级别提供了模块化和作用域，隔离性好。Go语言支持package和局部两种作用域类型。

#### package：

大写开头的是Public的。

小写开头的是Private的。

#### 局部：（函数或块内）

## 编程模型：

### C语言：

C语言支持过程式、函数式编程模型。

#### 过程式：

C语言通过函数来实现过程式编程。

#### 函数式：

C语言虽然可以通过函数指针来实现函数式编程，但函数指针是类型不安全的且这种实现语法古怪。

### Go语言：

Go语言支持过程式，函数式，面向对象的编程模式。

#### 过程式：

Go语言通过函数来实现过程式编程。

#### 函数式：

在Go语言中函数可以作为值和类型。

#### 面向对象：

Go语言的面向对象不同于C++，C#，Java等语言的面向对象，要比这些语言的面向对象简单很多。

#### 继承：

Go语言通过聚合和嵌入的方式来实现，如数据的聚合和嵌入和接口的嵌入。

#### 封装：

Go语言通过Method来实现封装（数据+函数）。

**多态：**

Go语言通过Interface来实现多态。