# Go Runtime

<https://zhuanlan.zhihu.com/p/111370792>

golang 的 runtime 核心功能包括以下内容:

1. **协程(goroutine)调度(并发调度模型)**
2. **垃圾回收(GC)**
3. **内存分配**
4. 使得 golang 可以支持如 pprof、trace、race 的检测
5. 支持 golang 的内置类型 channel、map、slice、string等的实现
6. 等等

下图是 golang 程序、runtime、可执行文件与操作系统之间的关系. 区别于 Java 需要安装虚拟机，go 语言的可执行文件已经包含了 golang 的 runtime，它为用户的 go 程序提供协程调度、内存分配、垃圾回收等功能.此外还会与系统内核进行交互，从而真正的利用好 CPU 等资源.

go 在处理协程上，使用了 GPM 调度模型，从而支持高效的并发调度. 如下图，内核线程与逻辑处理器是多对多的关系即 M:N. 从而提升并发效率. GPM 各个模块的解释如下:

* **G:** 即 Goroutine，更轻量级的线程，保存着上下文信息.
* **P:** Processor，是逻辑处理器. 将 goroutine 绑定逻辑处理器 P 的本地队列后，才会被调度. Processor 提供了相关的执行环境(Context)，如内存分配状态(mcache)，任务队列(G)等
* **M:** 它才是真正的计算资源，是系统线程.
* **全局队列（Global Run Queue）:** 未分配 Processor 的 Goroutine 保存在全局队列中. Processor 或 M 都可以从全局队列中取出 G .
* **本地队列（Local Run Queue）:** 是 Processor 的队列，当队列为空时，会从全局队列或其它队列补充 Goroutine.
* **sysmon 协程:** go runtime 会创建一个 sysmon 协程. 它会定期唤醒检查 goroutine 和 processor，确保 goroutine 不会长期占用 CPU 以及 Processor 可以被执行.

# Go为何没有go语言虚拟机？

Golang 非常适合服务中间件领域。

<https://www.zhihu.com/question/267065727>

# 进程，线程，协程

进程拥有自己独立的堆和栈，既不共享堆，亦不共享栈，进程由操作系统调度。（全局变量保存在堆中，局部变量及函数保存在栈中）

线程拥有自己独立的栈和共享的堆，共享堆，不共享栈，线程亦由操作系统调度(标准线程是这样的)。

协程和线程一样共享堆，不共享栈，协程由程序员在协程的代码里显示调度。

一个应用程序一般对应一个进程，一个进程一般有一个主线程，还有若干个辅助线程，线程之间是平行运行的，在线程里面可以开启协程，让程序在特定的时间内运行。

协程和线程的区别是：协程避免了无意义的调度，由此可以提高性能，但也因此，程序员必须自己承担调度的责任，同时，协程也失去了标准线程使用多CPU的能力。

原文链接：<https://blog.csdn.net/weixin_42117918/article/details/82025391>

# golang完全静态编译

<https://www.liangzl.com/get-article-detail-165139.html>

golang编译后的文件通常较大，依赖文件有glibc, ld等，为了简化分发，决定：1、将编译成独立一体的可执行文件，2、减少体积。

测试用例，reader.go（内容不重要，甚至包括了cgo）

go build reader.go

ldd reader

发现有不少依赖包，在整体打包前，还需要确认是否安装了glibc-static，否则会出错

# dnf --enablerepo=PowerTools install glibc-static

# go build -ldflags '-linkmode "external" -extldflags "-static" -w -s' reader.go

# upx -9 reader

1. 设置编译参数-ldflags '-linkmode "external" -extldflags "-static"‘，实现整体打包；
2. 设置编译参数-ldflags "-w -s"。其中-w为去掉调试信息（无法使用gdb调试），-s为去掉符号表，可有效减少编译后体积（一般能缩小30+%）；
3. upx -9，直接将3MB多的文件压倒800KB。（upx需要独立安装）

在独立打包的过程中，出现上述错误，经过查找资料发现操作系统少了glibc-static包造成的

直接使用yum安装： yum install glibc-static -y

安装后再次执行上述命令已经不报错了。

然后继续使用ldd命令检查，发现已经没有动态链接库依赖。