

1.3 PHP的生命周期

一个PHP实例，无论通过http请求调用的，还是从命令行启动的，都会向我们上一节说的那样，依次进行Module init、Request init、Request Shutdown、Module shutdown四个过程，当然之间还会执行脚本自己的逻辑。

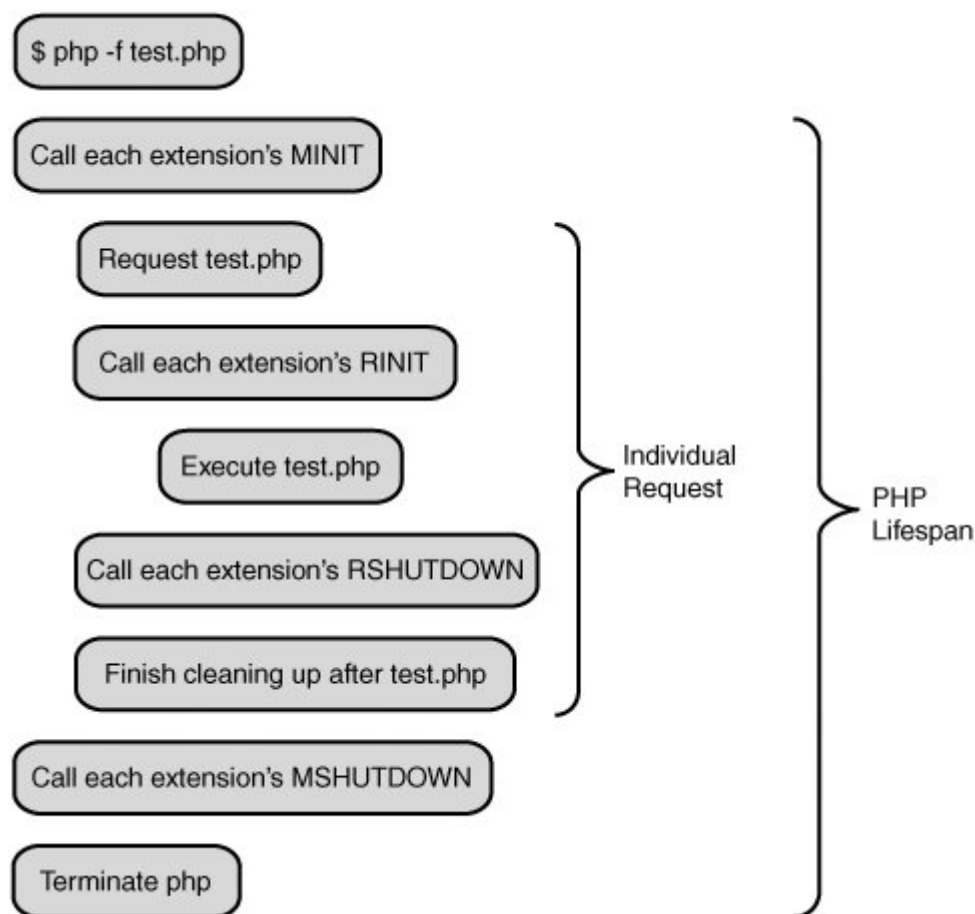
那么两种init和两种shutdown各会执行多少次、各自的执行频率有多少呢？

这取决于PHP是用什么sapi与宿主通信的。最常见的四种方式如下所列：

- 直接以CLI/CGI模式调用
- 多进程模式
- 多线程模式
- Embedded(嵌入式，在自己的C程序中调用Zend Engine)

1、CLI/CGI

CLI和CGI的SAPI是相当特殊的，因为这时PHP的生命周期完全在一个单独的请求中完成。虽然简单，不过我们以前提过的两种init和两种shutdown仍然都会被执行。图1.1展示了PHP在这种模式下是怎么工作的。



2、多进程模式

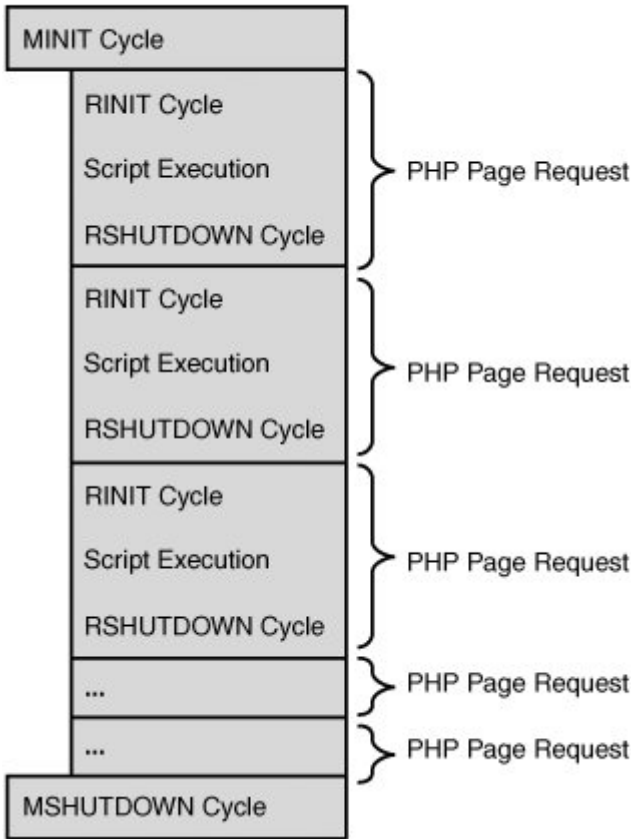
[ps:书是2006年出版的，所以你应该理解作者说多进程是主流]

PHP最常见的工作方式便是编译成为Apache2 的Pre-fork MPM或者Apache1 的APXS 模式，其它web服务器也大多用相同的方式工作，在本书后面，把这种方式统一叫做多进程方式。

给它起这个名字是有原因的，不是随便拍拍屁股拍拍脑袋定下来的。

当Apache启动的时候，会立即把自己fork出好几个子进程，每一个进程都有自己独立的内存空间，也就代表了有自己独立的变量、函数等。在每个进程里的PHP的工作方式如下图所示：

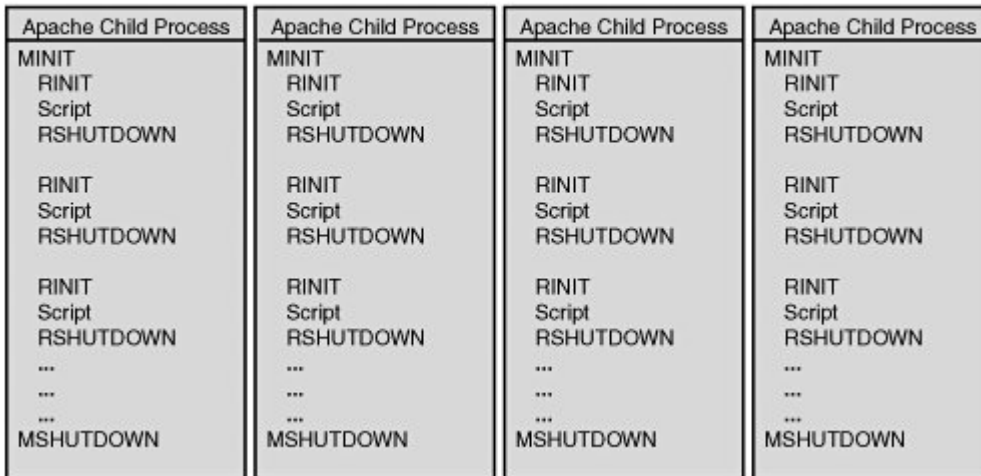
Individual Apache Child Process



因为是fork出来的，所以各个进程间的数据是彼此独立，不会受到外界的干扰(ps: fork后可以用管道等方式实现进程间通信)。

这是一片独立天地，它允许每个子进程做任何事情，玩七十码、躲猫猫都没人管，办公室拿砍刀玩自杀也没事，下图展示了从apache的视角来看多进程工作模式下的PHP：

Multiprocess
Apache
Webserver



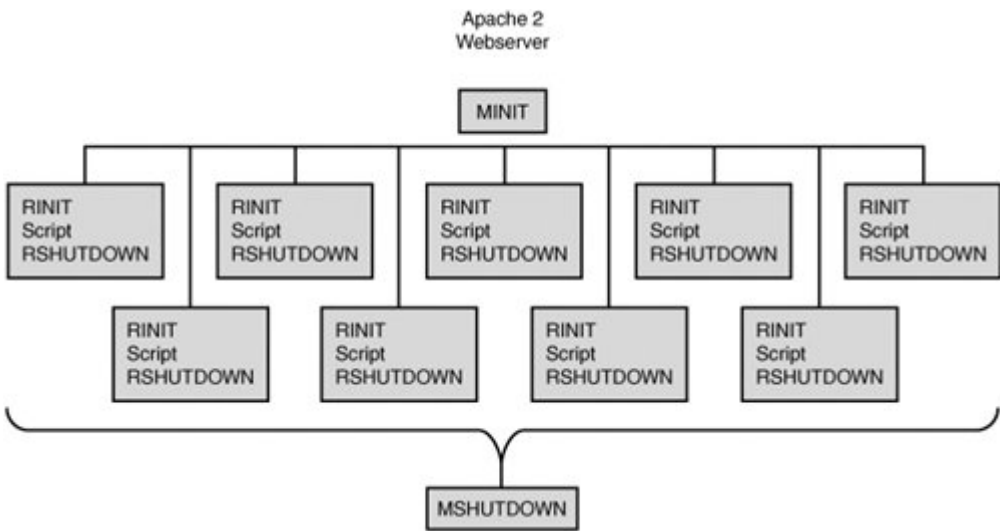
3、多线程模式

随着时代的进步，PHP越来越多的在多线程模式下工作，就像IIS的isapi和Apache MPM worker(支持混合的多线程多进程的多路处理模块)。

在这种模式下，只有一个服务器进程在运行着，但会同时运行很多线程，这样可以减少一些资源开销，像Module init和Module shutdown就只需要运行一次就行了，一些全局变量也只需要初始化一次，因为线程独具的特质，使得各个请求之间方便的共享一些数据成为可能。

其实多线程与MINIT、MSHUTDOWN只执行一次并没有什么联系，多进程模式下一样可以实现。

下图展示了在这种模式下PHP的工作流程：



4、Embed

Embed SAPI是一种比较特殊的sapi，容许你在C/C++语言中调用PHP/ZE提供的函数。

并且这种sapi和上面的三种一样，按Module Init、Request Init、Rshutdown、mshutdown的流程执行着。

当然，这只是其中一种情况。因为特定的应用有自己特殊的需求，只是在处理PHP脚本这个环节基本一致。

真正令embed模式独特的是因为它可能随时嵌入到某个程序里面去(比如你的test.exe里), 然后被当作脚本的一部分在一个请求的时候执行。控制权在PHP和原程序间来回传递。关于嵌入式的PHP在第20章会有应用, 到时我们再用实例介绍这个不经常使用的sapi。

关于Embed SAPI应用的文章

- [Larurence大哥的使用PHP Embed SAPI实现Opcodes查看器](#)

links

- [目录](#)
- 上一节 [PHP的启动与终止](#)
 - 下一节 [线程安全](#)