## 进程?

操作系统当中一个**进程**，就是一个正在运行的程序

Eg php aa.php 相当于创建了一个进程，这个进程会在系统中贮存，申请他自己的内存空间，系统资源，并且运行相应的程序

### 进程核心

进程核心1 内存：

进程核心2 上下文环境：

### 多进程

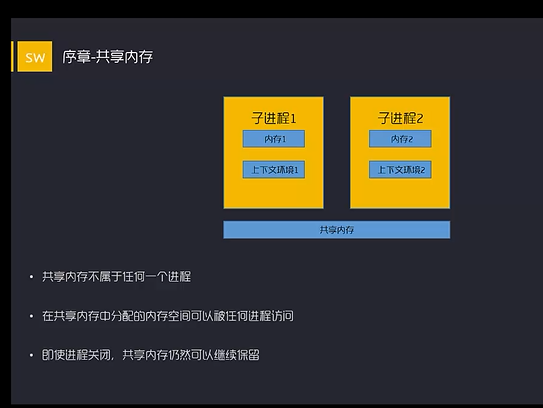
\*\* 复制 **主进程的内存和上下文环境**，子进程中变量修改不会影响父进程中的变量



### 进程间通讯?

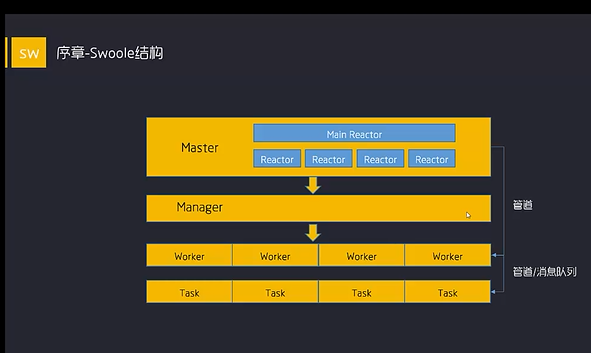
–共享内存（其中一种通讯方式）

不属于任何一个进程，任何进程都可以读取、修改



通过几个命令看一下 系统当中共享内存的分片

Ipcs –m



Web 依赖nginx php-fpm

Fpm 多进程php解析器

系统开销 创建和销毁进程，导致响应效率不是特别高

## Swoole架构（可以分为三层）

Swlooe采用了和fpm完全不同的架构

### （第一层）Master进程

用于处理swoole核心的事件驱动，拥有若干reactor线程

每个reactor线程用于，每个线程拥有epol函数的实例，对事件监听，在reactor中实现

A来自客户端的链接

B本地通讯用的管道

C异步操作用的文件描述符

链接、管道、文件

### （第二层）Manager 管理进程—仅仅只做进程管理和分配

创建管理下级进程（work、task）

### （第三层）工作进程a&b

a Work 进程：主逻辑进程，用于处理来自客户端的请求

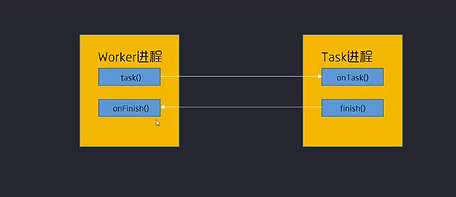
b Taskwork进程：异步工作进程，主要处理 耗时较长的同步任务

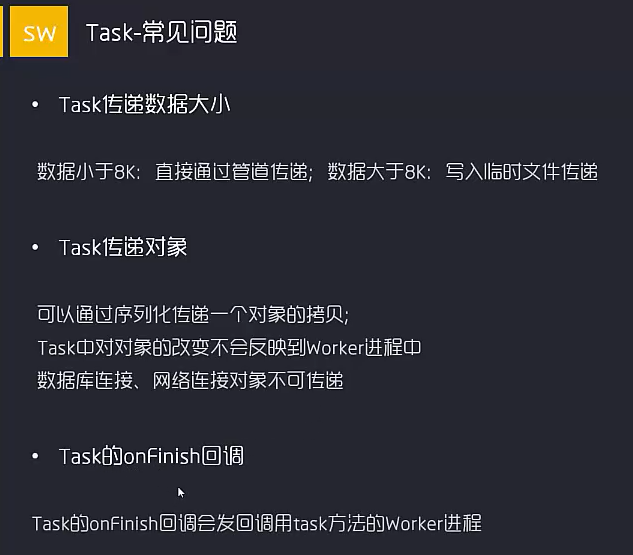
## 进程间通信 – 基于管道

可以配置，使通信走系统的消息队列

OnTask回调

Finsh函数





# Work、Task进程

Task进程是swoole当中独立于work进程的工作进程，用于处理一些耗时较长的逻辑。这些逻辑在task进程中处理时，并不影响work进程去处理来自客户端的请求，由此，大大提高了swoole扩展

Timer

版本1.7以前 线程

Ep 的回调timeout机制 堆 最小堆 剩余时间

堆算法

使用

永久定时器 tick

1. After