进程：一个可以独立运行的程序单位，例如Windows上的exe程序，Linux上的可执行程序貌似没有后缀。

线程：操作系统进行运算调度的最小单位。

进程就是线程的集合，一个进程有1个或多个线程组成。

Web服务处理request请求一般不会用多线程，一是因为多线程虽然是异步的，但不代表几个线程是同时运行的，实际上是系统在各个线程间来回切换，显然不符合多个request的要求，多个request理应不需要相互等待，都是独立的；二是多线程情况下，如果一个线程挂了，很可能导致其它线程都出问题，这也不符合多个request的要求。

单进程单线程：一个人在一个桌子上吃菜。

单进程多线程：多个人在同一个桌子上一起吃菜。

多进程单线程：多个人每个人在自己的桌子上吃菜。

从上面的描述上看，显然web服务最好的就是多进程单线程。

多进程开销大，多线程相对开销小。

线程安全：一般说来，一个函数被称为线程安全的，当且仅当被多个并发线程反复调用时，它会一直产生正确的结果。

要保证线程安全，属于同一进程的不同进程会共享进程内存空间中的全局区和堆，而私有的线程空间主要是栈和寄存器。同一进程的不同线程，函数的局部变量都是每个线程私有的，而全局变量，局部静态变量，分配在堆上的变量都是共享的，对这些变量访问时，必须加锁。

可重入：根据经验，所谓“重入”，常见的情况是，程序执行到某个函数foo()时，收到信号，于是暂停目前正在执行的函数，转到信号处理函数，而这个信号处理函数的执行过程中，又恰恰也会进入到刚刚执行的函数foo()，这样便发生了所谓的重入。此时如果foo()能够正确的运行，而且处理完成后，之前暂停的foo()也能够正确运行，则说明它是可重入的。

1. 可重入的函数不一定线程安全
2. 线程安全也不一定可重入
3. 不可重入的函数一定不是线程安全的。

预防不可重入：

（1）不要使用static变量和全局变量，坚持只用局部变量；

（2）若必须访问全局变量，利用互斥信号量来保护全局变量；

（3）不调用其它任何不可重入的函数；

（4）谨慎使用堆内存malloc/new。